Journal of Research and Practice in Information Technology

Special Issue:
A Tribute to John Bennett

Volume
33
number 4 2001

JOURNAL OF RESEARCH AND PRACTICE IN INFORMATION TECHNOLOGY
Formerly the Australian Computer Journal

The *Journal of Research and Practice in Information Technology* is a quarterly publication emphasising activities that have successfully connected fundamental and applied research in Information Technology with practical application. The journal publishes papers relating to both emerging research and to professional practice and thus contains articles that are of interest both to practicing information technology professionals and to university and industry researchers.

Editor (and address for correspondence):
Professor John F. Roddick
Editor-in-Chief
Journal of Research and Practice in Information Technology
Flinders University of South Australia
GPO Box 2100
Adelaide 5001, South Australia
Email: jrpit@cs.flinders.edu.au

Associate Editors
Associate Professor Graham Low
Associate Editor – Information Systems
The University of New South Wales

Professor Maria Orlowska
Associate Editor – Database Systems
The University of Queensland

Professor Gerald Quirchmayr
Associate Editor – Electronic Commerce
University of Vienna, Austria

Professor Bernhard Thalheim
Associate Editor – Conceptual Modelling
Brandenburg Technical University at Cottbus, Germany.

Dr Hugh Williams
Associate Editor – Computer Science
RMIT University

Denis Warne
Associate Editor – Software Engineering
BAE SYSTEMS

Michael Oudshoorn
Books Editor
The University of Adelaide

Professor Ian Witten
Associate Editor – Digital Libraries
University of Waikato, New Zealand.

Tom Worthington
Director, Publications Board
Australian National University

Office Bearers
J. Ridge, President
R. Kalkman, Vice President
P. Argy, Vice President
P. Ralston, Immediate Past President
R.G. Heinrich, National Treasurer
D.W. Furini, Chief Executive
PO Box Q534, QVB Post Office, Sydney, NSW 1230, Australia
Telephone: (02) 9299 3666
Facsimile: (02) 9299 3997
email: info@acs.org.au
URL: www.acs.org.au

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT's copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by the Australian Computer Society Inc.

Print Post Approved PP255003/01660

John Makepeace Bennett – An Inspiration

Jenny Edwards
Faculty of Information Technology, University of Technology, Sydney
jenny@it.uts.edu.au

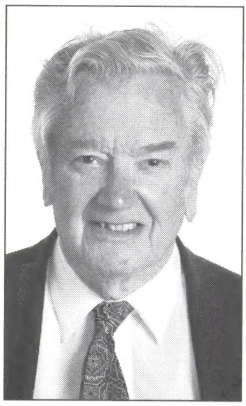
This volume is a celebration for John Bennett's 80th birthday. It was written with respect and affection from just a few of the many he has inspired over his long professional career. It also commemorates indirectly, 40 years of the Basser Department of Computer Science at the University of Sydney, of which he was the first Professor and Head.

Keywords: John Makepeace Bennett, Basser Department of Computer Science, computing pioneer
CR Categories: K.2 – History of Computing

1. INTRODUCTION

This paper does not purport to be a biography or even a summary of John Bennett's (hereafter referred to as he himself does, as JMB) distinguished career. Rather it illustrates some of his many interests in computing and the work they inspired in others. Those readers desiring more details may find them in Bennett (1990; 1994; 1998).

While taking a leading role in the inception of the Australian Computer Society (ACS), of which he was the first President, he was also a major player in the development of the Australian Computer Journal, now known as the Journal of Research and Practice in IT. Thus it is fitting that this festschrift forms a special edition of the journal. As the papers will show, especially that of de Ferranti* and Deane*, putting research into practice has defined JMB's life.



2. EARLY YEARS

John Makepeace Bennett was born in Queensland in 1921. After completing his schooling he graduated with a Bachelor of Civil Engineering from the University of Queensland. He then joined the Royal Australian Air Force being posted to a radar unit. Ironically some of his lectures were held in buildings at the University of Sydney which were later to be inhabited by Basser. As Barry de Ferranti* recalls,

> *we had severally been students at the Physics School and received the benefit of Professor V A Bailey, though at different times. 'Rusty' Bennett was one of the WWII 'Bailey Boys', a much respected Radar trainee group, which still meets*

(Names marked with an * are authors in this volume.)

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

in reunion. In the 40s Bailey and the other Mathematics and Physics staff had provided renowned lectures to Service personnel and then after the war to us BSc types, on subjects such as radar antennae design.

Inspired by a radio lecture, in September 1947 JMB arrived at Cambridge to do his PhD under one of the pioneers of computing, the still active, Professor Maurice Wilkes.

> John Bennett was my first research student. He joined the Mathematical Laboratory, as the Computer Laboratory was then called, in 1947. I first heard his cheery voice when he called me up to say that his boat from Australia was late arriving and that he had only just landed. He described himself as our erring Queenslander. He soon made up for being a week or so late, and made a great contribution to the work and life of the laboratory. In particular, he pioneered the use of digital computers for X-ray crystallography. Over the years, I have seen a lot of John, both in the UK and in Sydney. It gives me pleasure to join with his other friends and admirers in congratulating him on his 80th birthday, and wishing him many more years of fruitful activity.
>
> **Maurice Wilkes**

At Cambridge JMB was involved in the construction of EDSAC, one of the first modern computers. On completion of his PhD he joined Ferranti, where he had a major role in designing their latest computer, the PERSEUS, intended for commercial data processing, particularly insurance applications, and he lead their programming team. During this time he visited the University of Illinois team just after they completed ILLIAC. The paper by Barry de Ferranti*(from the early SILLIAC days) and John Deane*(SILLIAC historian) illustrates the importance to Australian computing of the subsequent link between JMB and the ILLIAC.

This period also saw the start of a long marriage. John and Mary Bennett celebrate their fiftieth anniversary on Australia Day, 2002. While all four of their children have pursued professional careers, the eldest, their son Chris*, is the only one to have chosen technology.

3. BASSER YEARS

In 1956, JMB accepted a position in the School of Physics at the University of Sydney as the numerical analyst. He subsequently became a Professor and Head of the Basser Department of Computer Science, a position he held until his retirement in 1986. Since arriving at Basser, with which his name became synonymous, his enduring vision was to educate students, industry and government in the powers of computers – using whatever platforms were available.

In that 30 years his influence on both academic and industrial computing in Australia was widespread. Despite all the activity of teaching, building revolutionary networks and liaising with industry, JMB has always been an active researcher. He has over one hundred research articles and an eclectic range of interests, with his latest enthusiasm being quantum computing.

His abiding interest is matrices, illustrated here by a paper on Hadamard matrices from former staffer, Jennie Seberry* *et al.* The extension of JMB's work in matrices to mathematical programming is represented by Ailsa Land's* interesting application of network programming to an unusual problem.

One of the strengths of Basser was the quality of visitors JMB was able to attract. Among many others, early lecturers and researchers included Dr Trevor Pearcey, the developer of CSIRAC,

(Australia's first computer), Professor Maurice Wilkes from Cambridge, Professor Sandy Douglas, long associated with the British Computer Society, numerical analysts such as Professor Leslie Fox from Oxford and Professors Ailsa and Frank Land from London School of Economics. Ever mindful of his budget, in the Lands, JMB managed to attract a well known mathematical programmer and an information systems expert at the same time.

JMB's enthusiasm for matrices and mathematical programming took fruit in his early PhD students. He was an inspiring supervisor and instilled in this author, his second PhD student, (and I'm sure, many others) precision in research writing and the importance of asking the right question. His first PhD student was Robert Dakin, one of the pioneers of the branch and bound method and who has had a long career in academia and the computing industry. The following shows the confluence of many ideas stemming from his early numerical analytical work with JMB.

I want to thank John Bennett who planted some fascinating, still germinating seeds in my psyche.

Following my PhD under JMB on integer programming, one of my examiners, Ailsa Land suggested I do more work on the Travelling Salesman Problem (TSP) but alas all of my TSP work since then has been a sporadic, spare time, hobby, designed for PC solution and education and exploration rather than the record beating supercomputers of the latest literature. All of my test examples are symmetric TSPs with link lengths being euclidian distances between nodes on a plane. This was challenge enough, and had the advantage of allowing me to follow the progress of algorithms using graphic displays. All of the algorithms are readily extended to the general symmetric and asymmetric cases, but there are no performance results at this stage.

Variations

My starting point was the notion of building a tour sequentially, starting from a trivial tour of two (symmetric case) or three (asymmetric) nodes from the TSP and inserting nodes sequentially until a complete TSP tour has been constructed. The following ideas have been developed from this simple starting point.

1. Minimum increment

In this case the nodes are inserted one at a time, choosing the point of insertion to minimise the resulting increase in tour length. The resulting tours are not too far from optimal for small problems but become worse as the number of nodes increases. It is very fast, with computing times of $O(n^2)$ where n is the number of nodes.

2. Branch and bound

This is a conventional branch and bound algorithm in which the branching consists of choosing insertions with increasing increments. Each bound is simply the length of the current sub-tour, though they are not strong enough, resulting in lengthy searches. At least the solutions reached are truly optimal.

3. Optimal sub-algorithms

I found that one could obtain a true optimum solution within a subset of all possible tours using a dynamic programming approach. Initially the subset contained $3*2^{n-4}$ tours (2^{n-2} in the asymmetric case) and the solution time was of $O(n^2)$. Later I found ways of increasing the size of the subset at the cost of increased computing time which still was $O(n^2)$. One

Continued next page

Continued from page 275

could vary the size of the subset by varying the computational penalty. Another variant could be optimised in times of O(n). It needs to be fast, since it is invoked many times to process one TSP.

4. Tour improvement algorithm

This approach uses repeated application of optimal sub-algorithms to produce a series of tours with diminishing length. The algorithm provides an efficient way of finding what we might call a "good engineering solution". The snag is that you have no way of knowing how far it is from the true optimum.

An integer programming formulation

One of a sequence of insertions has major implications for later insertions. If node k is inserted between nodes i and j then the gap (i,j) is no longer available for later insertions and two new gaps – (i,k) and (k,j) are created and available for later insertions.

The omission of these effects places major limitations on the effectiveness of the minimum increment and branch and bound algorithms described above. The simplest way I could find to model this kind of interaction was an integer program which, in fact, includes the complete sequential insertion process.

The general form of the model is intriguingly similar to that of integer programming models of the Assignment problem and the Transport problem. Perhaps the vertices of the constraint polygon of the TSP model without integer constraints have integer coordinates, as in the other two models? Perhaps the TSP model is also amenable to efficient tailored solution algorithms?

More detail may be found at http://www.pcug.org.au/~dakin/tsp.htm.

Robert Dakin

Inspired by a question from one of JMB's visitors is the paper by Don Herbison-Evans* on the resulting thirty years of work on computerising dance notation. Ian Parkin*, another long serving Basser staffer, contributes a somewhat more theoretical graphics paper.

Perhaps owing to his early experiences in constructing computers, another of JMB's lifelong interests has been hardware. Another early PhD student, Chung Yuen* discusses memory consistency. The teaching hardware theme is illustrated at a hands-on level by Allan Bromley's* paper looking at the mini logic training devices which have been used for some thirty years at Basser.

From early times, JMB often said that he considered computing a perfect profession for women, offering flexible work conditions and no serious bars to participation. He undertook initiatives to encourage more women into computing throughout his professional career. He had some success at the local level, as perhaps shown by two former students Jenny Edwards* and Judy Kay*, both of whom have been active in computer education for many years, especially for women. However, as their paper concludes, the battle to encourage more women into computing education has not been won on the wider front.

JMB was always adept at acquiring the latest hardware only to stretch it way beyond its stated capabilities and purpose. De Ferranti* and Deane's* paper mentions Bassernet, one of the world's first LANs. Some of the early links of what was to become AARNET were run into Basser's building. This might be seen as a foundation for two practical and complementary papers discussing

somewhat more modern networks. Industry consultant, Chris Bennett* (JMB's son) looks at the problems of delivering streaming media via broadband over the Internet and Arthur Sale*, an early Basser staffer, examines cheap access to broadband for the masses.

Above all else JMB was an educator. It is fitting to finish with two research papers which inform practice. Jan Hext* another early staff member discusses B-trees and Norman Foo* a long term Basser staff member looks at a classical computing problem in nondeterminism, thus linking back to JMB's early days in the UK where he worked with the brilliant computer pioneer, Alan Turing.

One of JMB's greatest achievements is the proliferation of his staff and students in senior positions throughout academia and industry – too numerous to name, many were present at Basser in July 2001 to celebrate his birthday. Some of those who have served as senior academics are:

Jenny Edwards*	Professor and head of computing at UTS
Norman Foo*	Professor at Basser and UNSW
Mike Fry	Professor and Dean of the faculty at UTS
Jan Hext*	Professor and head of computing at Macquarie University
John Hughes	Professor and Dean of the faculty at UTS
Bob Kummerfeld	Associate Professor and head of Basser
Carroll Morgan	Oxford University and adjunct Professor at UNSW
Vaughan Pratt	Professor at Stanford and one of the founders of SUN
Ross Quinlan	Professor and head at UTS and Basser
John Rosenberg	Professor and head of Basser, Dean of the faculty at Monash University
Arthur Sale*	Professor, head of computing and Pro-Vice Chancellor at University of Tasmania
Jennie Seberry*	Professor and head of computing at Basser and University of Wollongong
David Skellern	Professor at Macquarie University
Chris Wallace	Professor and head of computing at Monash University
Chung Yuen*	Professor and head of computing at National University of Singapore

John Bennett was awarded an AO in 1983 for his visionary contributions towards the development of computing in Australia. These are eloquently summed up by Chris Wallace, an early Basser staffer.

Can you imagine giving a bunch of people who had never seen a computer an instruction manual covering fully both sides of one sheet of paper, and getting most of them to write, debug and run a program for calculating the present value of an annuity in about three hours? In a "language" like a really primitive assembler? On a machine with 5K bytes of memory? It is not something I would attempt with today's students, but I recall John Bennett doing this several times, while introducing computing to some of Sydney's businessmen. Somehow, he managed to infect others with his own conviction that it was all really very simple.

Continued next page

Continued from page 277

Of course, to him it really was. And so were the linear algebra, numerical methods, simplex algorithms, fast Fourier transforms, eigenvalue problems etc. which he expounded in his more formal lecture courses, but somehow to students like myself the simplicity seemed to recede as the matrices multiplied, leaving John sometimes mildly perplexed as to how we could find anything so transparently elegant at all difficult.

John remained an incurable optimist throughout his academic career, always seeing his students as a little better than they usually were. Always he hoped that the promise offered by computing would become clear to the powers that dictate our education, business and industrial policies. He argued his case forcefully through every channel available, and predicted remarkably well the impact this revolutionary tool would have on our society. Unfortunately, he could almost claim the mantle of Cassandra, blessed with the vision to see the future, but cursed to be disbelieved. He predicted twenty years ago the huge demand for computing skills, and did all he could to ensure the demand was met. Of course, it was not, either in quantity or quality, and we now suffer from those decades of neglect. He anticipated the power unleashed by letting computers communicate with each other, and led some of the earliest developments in high-speed computer links, but there were never enough resources or staff fully to achieve what he saw was possible.

In celebrating his 80th birthday, one rejoices at the great achievements of his career, and can only wonder what more he might have done had he ever had the support and resources he merited.

Chris Wallace.

It is fitting to finish with words from the toast given by Jennie Seberry* on his retirement:

Samuel Johnson in *The Vanity of Human Wishes* said

"To talk in public, to think in solitude, to read and to hear, to enquire and answer inquiries, is the business of scholars."

Scholarship and imparting knowledge is John's business.

Footnote: 2001 sees the end of an era. On 1 January 2002, the Basser Department of Computer Science at the University of Sydney becomes the School of Information Technologies.

REFERENCES

BENNETT, J. (1990): Early computer days in Britain and Australia - some autobiographical snippets. *Annals of the History of Computing* 12(4): 281-285.

BENNETT, J., BROOMHAM, R., MURTON, P., PEARCEY, T. and RUTLEDGE, R. (1994): *Computing in Australia - the development of a profession*. Hale and Iremonger.

BENNETT, J. (1998): Autobiographical Snippets. *In In the beginning - personal recollections of software pioneers*. GLASS, R.(ed). IEEE Computer Society Press: 212-237.

BIOGRAPHICAL NOTES

Jenny Edwards gained a BSc, MSc and PhD, all in the Basser Department of Computer Science at the University of Sydney. She was John Bennett's second PhD student. After working at Basser for some years, she joined what is now the Faculty of IT at the University of Technology, Sydney, where she is a Professor and was Head of School of Computing Sciences from 1994-2000.

Her research interests include mathematical programming and algorithms for parallel processing. She is a Fellow of ACS, and a member of SIGCSE, ASOR, INFORMS and the Mathematical Programming Society and has served on their committees for many years. She has also been active in various Women in Computing initiatives and participates in FITT activities.

Early Contributions to IT in Australia,

from the Basser Department of Computer Science – John Bennett's SILLIAC years

B. de Ferranti

Barry Ferranti Associates ®, Woollahra, Australia
bferranti@cs.su.oz.au

J. Deane

CSIRO Telecommunications and Industrial Physics
Radiophysics Laboratory
PO Box 76, Epping NSW 1710, Australia
john.deane@csiro.au

Some important beginnings of what became the Australian IT Industry can be found in the way the Basser group went about teaching those from government and industry the rudiments of automatic computation and data processing with 'hands-on' experience. The early contributions of John Bennett and his colleagues enabled many who attended courses, visited and used SILLIAC, to examine the potential of IT. Many went on to make decisions for their organisations which would have long-term benefit and impact on the growth of this country as an effective and efficient user of the emerging technology.

Keywords: SILLIAC, UK, US and Australian computing; emerging IT; commercial applications of computing.

CR Categories: K.2 – History of Computing.

1. BACKGROUND

The setting for IT in Australia

The first half of the twentieth century saw Australia well astride the sheep's back but looking around for other industries on which to build a strong future. Information Technology (IT) had not yet appeared, but scientific computation and commercial data processing were in need of automation, and the campus of the University of Sydney became a centre of development (research and practice).

Data processing was done mainly by hand on calculators or through punch-card/tabulator installations. As with other capital equipment, the major source for such business or scientific equipment and associated application techniques, was the manufacturing industry of the UK. Few examples of US techniques were to be found in the regular government or industry contexts – that would come soon enough.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

Innovation and ingenuity were not yet key words in the politicians' lexicon yet the seeds of technological prowess were there in various forms. Approaching the mid 20th century the political vocabulary was more likely to embrace minerals, agriculture and, particularly, drought.

The totalisator was an outstanding Australian invention. Sir George Julius' contribution to the world's wagering industry was a harbinger of what special-purpose automatic computing might achieve. That was pre-1939. Even so, post WWII, the nation's major engineering investment was aimed at irrigation, power, and civil works.

The Snowy Mountains Scheme demanded extensive computation, analogue and digital. That spawned another special purpose computational effort in the CSIRO Mathematical Instruments Section in the Electrical Engineering Department at The University of Sydney. Then came a huge opportunity. Trevor Pearcey and his CSIRO team were developing and constructing one of the world's first electronic digital general-purpose computers, later called CSIRAC. Instead of capitalising on this achievement the Australian government chose to fund 'rain-making' research instead! Despite this, the climate was right for further development of scientific computing.

Meanwhile the 'town' was demanding data processing and the 'punch-card' systems providers sought to extend their capabilities with the emerging electronics, offering, or at least promising, speed and precision.

So, in 1950, the setting for what was to become IT in Australia, and its effective use by Government and Industry, comprised: the Snowy Scheme, civil, mechanical, electrical, communication and control engineering, CSIRO and CSIRAC, the Universities, and the rich legacy from WW II – people skilled in radar and electronics.

Mathematics, Physics and Engineering

Suitable careers for raw graduates in the hard technologies were few and far between in Australia. Consequently many went overseas for 'experience', mostly to England. They used their Sydney University studies in mathematics, physics and engineering to participate in leading-edge research and development as part of their time in the UK.

In the early 1950s, a new head of the school of Physics, Harry Messel, set about lifting its image, and fortunes. He attracted staff from overseas, including several who comprised a strong cosmic ray research group. He also convinced donors that this 'nuclear physics' was worth funding and secured generous capital. Essential to their work and the associated particle physics was a high performance computer. A Sydney development of the University of Illinois' machine, ILLIAC, was deemed appropriate, initial capital was attracted and manufacture of component units for the machine was contracted out in July '54.

2. SILLIAC GESTATION

Building SILLIAC and planning its use

In June, 1954 Harry Messel hired Brian Swire (an early user of CSIRAC) as chief engineer to build SILLIAC (Sydney version of ILLIAC). De Ferranti was hired to assist Swire, with specific responsibility to accept units subcontracted to STC, assemble and test the system, and to participate in its development. He returned to Australia from the UK via Illinois, where the progressive development of ILLIAC was proceeding. There was an emphasis on speed and reliability, which Meagher, Robinson and their team had built into the ILLIAC design, which would prove beneficial in SILLIAC. Their enthusiasm was contagious and their generosity in handing over the program library and their experience, enormously valuable.

American Precedents

In designing ILLIAC, the University of Illinois took the Von Neumann (Princeton) approach, and went for speed through parallel arithmetic operations in a re-engineering of his IAS computer. The notorious unreliability of earlier electronic calculators, like ENIAC, had left bad impressions of the potential of electronic stored program machines. In the US the acceptance climate set by business and defence demands was rigorous, and that was to be to SILLIAC's advantage. Although IBM and others were wary at first, business machines would soon follow.

US-UK-Australian Milieu

The School of Physics now relatively well funded environment encouraged it to take the best of all overseas development and build a centre of computing excellence. ILLIAC's link with Cambridge UK, through Wheeler's operating system, programming and documentation, meant that the SILLIAC team would have a flying start in computer application.

Basser Benefit

Funding SILLIAC had taken a typically Australian flavour. Among Messel's contacts was Adolph Basser who besides being a jeweller, raced horses in such events as the Melbourne Cup. The story is that his winnings were donated to the *Nuclear Research Foundation* (now the *Science Foundation for Physics*) to build the computer laboratory – not only initially but a second time, permitting the installation to expand – an eponymous augury.

It was with these many advantages that the physical system of the computer was constructed in the rooms adjacent to the main entrance to the Physics Building at Sydney University. 'Basser' was quickly taking shape. What was still needed was the programming effort and direction to bring on the applications.

By late 1955 most of the machine was built but they still needed someone with wide application experience to guide future users.

Imperial College Course

In June 1955 the computer group at Imperial College (London) ran a course for UK industry. To this was sent one of the authors (de Ferranti) from the GEC Research Lab, where he was working in pulse communications and reviewing potential applications for computers.

The course was extensive and included current and proposed applications, and visits to scientific and commercial installations. Opportunities to apply any of this back in Australia seemed only remotely possible.

Ferranti Limited sent two top representatives to the wrap-up of this course – Chris Wilson and John Bennett. They came to give the commercial view, but enlarged the picture to project where computer-based data processing was going. Bennett's wide experience of English computing development and application gave invaluable insights, including the possibility of specialised designs, e.g. for the insurance industry. Clearly here were computer experts who understood and welcomed the demands of commerce. It was impressive that an Australian had already done so much to be a broadly-based computer guru at that early stage of what was to be the IT industry.

Thus the Nuclear Foundation's newsletter, Nucleus (1955) announced in November that they had found their guide and Numerical Analyst:

> *We are happy to announce that we have been able to lure a first-rate Australian away from a first-rate job in England to join our SILLIAC team.*

3. SILLIAC'S FIRST YEAR – TEAMWORK

John Bennett was installed in February 1956, and had hardly set foot back in Australia when he found he was to present SILLIAC to the public for an Open Day. His talk included a general introduction then a status report:

> *... Both the construction and testing of this machine are well advanced, and although today we will not be able to show you a complete working machine, we will be able to give you a demonstration of the arithmetic sections of the machine, and of the paper tape preparation equipment...*

He took a wide view:

> *... it is clear that the responsibility of the Adolph Basser Research Laboratory should extend beyond the mere provision of a tool for nuclear research. In fact we consider that we have a prime duty to help members of the community who are likely to be affected by these new techniques...*

Specifically they would run classes and provide access to the computer for people outside the University of Sydney – especially users from industry, banks and government (who would be able to pay the 50 guineas fee) (Bennett, 1956a).

An enthusiastic prospective user was the Snowy Mountains Hydro-Electric Authority (SMHA) which had been working on Australia's biggest ever engineering project since 1949. Immediately following the Open Day, John Bennett was taken to SMHA HQ at Cooma where he was given a tour and a grilling. They had money, and huge computing problems which ranged from tunnel diameter optimisation, dam construction modelling, fitting survey data and general statistics to a requirement for some sort of model to let them develop operating procedures for all the dams, flows and electricity generating turbines. Following four days of intense meetings the SMHA decided to try to hire Bennett as a consultant, train one of their engineers for SILLIAC, prepare a trial problem, investigate options for their own computing equipment, and as soon as possible use SILLIAC for survey data correction (Bennett, 1956b).

Shortly after this, Bennett organised Basser's first computing course and could advertise through the Institution of Engineers, Australia.

> ***NEWS ITEM***
> ***Courses on Digital Computing***
> ***(Including the Use of the SILLIAC)***
> *The Nuclear Research Foundation within the University of Sydney announces a course in the Adolph Basser Computing Laboratory on Digital Computing Techniques. This course is designed to introduce graduates in Science and Engineering to the techniques required for using high speed automatic computers.*

A good understanding of commercial practice was demonstrated by simultaneously announcing the same course to the Institute of Accountants with:

> *This course should be of interest to those working in the field of engineering, the physical sciences, statistics and economics, and to more mathematically-minded accountants.*
> (Bennett, 1956c)

This was not just a programming course as users also had to be introduced to numerical analysis to get useable results from the machine.

Computing service to all disciplines, was to be the essence of the Basser approach. Others would be helped to organise and program their solutions to numerical problems, and the team worked with them to learn more about systematising their computation. Mutual problem solving with experts, staff, users and the resources at hand became a stimulating pursuit of excellence. The views of all comers were welcomed on this voyage of discovery.

With the hardware visibly taking shape another John Bennett responsibility triggered a letter to the head of the University of Illinois Computer Laboratory, Dr Ralph Meagher:

> *...You may recall that I visited your Laboratory towards the end of 1953, and you probably know that I have now joined Professor Messel's group to work on the SILLIAC.*
> *One of my first problems is to get a hand-book out, and clearly the most sensible thing for us to do is to plagiarise yours, as far as possible, particularly as the matter is becoming rather urgent. I wonder if you are in agreement with this? ... We will of course give the appropriate acknowledgment.* (Bennett, 1956d)

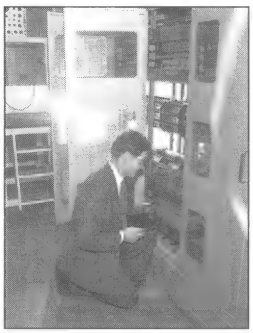


Figure 1: The senior numerical analyst, Dr JM Bennett inspects SILLIAC's complex wiring system at Sydney University's School of Physics, 1956 (National Archives, 2001)

This was agreed, and the suitably modified text was produced as the *SILLIAC Programming Manual* in June (Basser, 1956a). As soon as a copy of the ILLIAC operating and programming manuals arrived, the team encouraged users, not only physicists but anyone else on the campus, who might reasonably make use of SILLIAC, to think about programming the as yet incomplete machine.

In the event it was a propitious start when the machine ran its first program on 4th July, 1956 – a computation on the 'three body problem' of nuclear physics, for Dr John Blatt, an American who had worked on ILLIAC. An interesting episode, soon after, was again related to horse racing! Dr Peter Claringbold of Veterinary Physiology, came, took away a copy of the manual and soon returned with a, for then, very large program which would just about fill the available space in the 1024x40 bit memory. Bennett in particular, but really all who had done any programming, warned that such a maiden effort was unlikely to work. To all round amazement, Claringbold's program ran, first time. The application? – Correlation of race horse winnings with physiological characteristics, in particular heart size. Perhaps this influenced a career change as he later headed the CSIRO Computing Division and played an active role in Australian professional computing societies.

Courses

The courses to introduce Industry and Government to modest data processing proved a steep learning curve for everyone. Finding pertinent applications to demonstrate and to set as exercises was a challenge overcome on the run. The emphasis was on attracting corporate business and government, and to explore with them the potential for application in their contexts. For example, de Ferranti produced a program showing how a bank manager might automatically check the tendencies of customers towards overdraft, as a decision aid.

Brochure

Promotion of the Basser Laboratory included suggesting applications and means of addressing them. Bennett advised de Ferranti on what might be interesting and useful avenues for potential users and the vehicle for publicising this, *The Electronic Computer "SILLIAC" and the Adolph Basser Computing Laboratory* (de Ferranti, 1956), was produced as a 20 page brochure, detailing history and 'How to Make Use of SILLIAC', and distributed in advance of the official opening.

4. REACHING OUT

Telling the World

The already multi-national heritage, of Basser Lab's UK and US contacts, stimulated visits by many interesting figures, even before the first program was run. By May 1956 there was discussion with significant local prospects, such as AGL to help solve gas-pipe network problems. Bearing in mind that the Cold War was still well and truly on, SILLIAC was neutral territory for visitors from such unexpected places as the Soviet Union and China. Bennett was ever the open handed diplomat in discussing, explaining and promoting the abilities of this new technology. Every opportunity was taken to widen the window of application opportunity.

Opening

Nuclear Research Foundation people flocked to the official opening, 12 September, 1956. The level of understanding of numerical analysis might not have been high but enthusiasm and the obvious competence of the Basser team seemed to engender interest and, more significantly, a desire to do business with it. Bennett lost no opportunity on this and subsequent occasions to nail their needs. Among Bennett's contributions to the SILLIAC Opening Ceremony on 12 September 1956 was a multi-player game of NIM. He had previous experience here as he had been involved in initiating and demonstrating Ferranti's *NIMROD*, a purpose built game machine (the first?) which was designed to demonstrate computer principles (Bennett, 2001).

Demonstration

Banking and more technical computing examples were devised to demonstrate the system (and staff) prowess and steadily a schedule of paying applications was built up. A whole shift of work was within sight by year's end. The SILLIAC team were soon being consulted on diverse problems and the commercial as well as scientific success of Basser seemed assured.

5. OUTWORKINGS

Business and Government take notice

A miscellany of outside user applications was analysed, coded and run in the first few months. Some examples, most of them requiring considerable assistance from Bennett and the team, follow.

Woolworths: The company secretary/accountant came on a SILLIAC course and then tested his ideas of inventory control, using the consulting service.

AGL: The structural analysis technique known as 'Hardy-Cross' was successfully adapted to pinpointing leaks in the Sydney domestic gas supply pipe network.

PMG (both telecommunications and post office): Several areas, ranging from telephone traffic to payroll were tackled. Of these perhaps the longest running, and lucrative, was the 'Unofficial Postmasters Payroll' – pure commercial data processing.

There were enquiries from Qantas Empire Airways Ltd, Standard Telephones and Cables, and

the CSIRO Division of Animal Genetics, and Bennett (1956e) provided an enthusiastic update to the Snowy Mountains Hydro-electric Authority. The ensuing projects reported in the technical journals started with a swag of items in the 1957 conference on *Data Processing & Automatic Computing Machines* held at the Weapons Research Establishment at Salisbury in South Australia. Basser sent eight delegates and the first section, Programming and Mathematics, included *The SILLIAC* – a grand introduction by Bennett and Swire(1957). Bennett(1957a) also presented *The Use of Continuants in Practical Numerical Analysis.*

Employment and Careers

In 1956 there was little public or official anticipation of the likely social impact of automatic computation. Significantly, the University pioneered with a forum for discussion of likely effects on employment and careers, which later led to productive cooperation with the Commonwealth Department of Labour and National Service (Australian Government, 1968).

Multinationals show interest

IBM was only recently established in Australia as a supplier of electronic data processing machines, but some of its customers, like AGL and the insurance companies, were ready to move to computers as soon as sensible plans for their application and staffing could be put in place. NCR, BTM (Hollerith), Kalamazoo (Powers) and others were all seeking to enter the emerging market. Despite the academic emphasis of the university, thanks to its broad spectrum philosophy, Basser played a pivotal role in introducing such suppliers and users to IT concepts.

On SILLIAC's first birthday the Department started a series of more-or-less quarterly reports - which are still strongly supported at the time of writing, over forty years later. The first was 'A Year's Operating Experience with SILLIAC'. In it Bennett (1957b) concluded

> *To me, perhaps one of the most interesting effects which has taken place over the last year has been that people from different departments with mathematically similar problems have been brought together in the Laboratory. Resulting discussion – once the difficulties of specialised "chapel-talk" have been overcome – has resulted in considerable cross-fertilisation. The nuclear physicist finds that he has a similar problem to the geneticist, and the veterinary scientist to the computer designer – and everyone profits from the discovery!*

Key Computing Techniques and Applications emerge

1955-56 was a pivotal period for IT in Australia. What had been an area of primarily academic interest, scientific numerical computation, was quickly converted to a focus of industrial and government involvement. For The University of Sydney it was particularly significant, in that the University campus, combining as it did the research groups of relevant faculties like Engineering and Science and the National Standards Laboratories of CSIRO, became a centre of excellence for automatic computing and data analysis.

Key to the rapid success of this focus was the Messel philosophy, to attract back to Australia expatriates, who had relevant overseas experience. The team he hired to build and apply SILLIAC comprised a group of engineers capable of looking outside the field of physics towards a computing and consulting service which would reach far beyond the solution of immediate problems of his School's disciplines.

Without question the experience and attitude which John Bennett brought to the Basser Lab, of attracting the attention of potential users of SILLIAC, then identifying methods and coaching user personnel to achieve practical results, stimulated a new era. This applied research approach created new techniques and revealed novel applications, which industry and government were able to put into practice at Basser and/or in their own installations, which followed.

From Technical Interest to Commercial Reality

The examples mentioned (Woolworths, Gas, Postmaster Payroll, Networks, etc.) were but a few of the applications which started with a curiosity approach, then matured through further technical development of the methodologies developed there at Basser. Bennett and his colleagues brought an indefatigable attitude to assisting all comers. Out of what might have been seen by some as purely academic or technical pursuits, came a spreading network of pursuers of computing solutions to every-day problems. Bennett's evangelical enthusiasm and competence helped bring IT commercial reality.

6. HARDWARE AND SOFTWARE DEVELOPMENTS

SILLIAC's magnetic tape system was being developed by Basser, and Bennett had responsibility for its software. He was keen to get practical advice and arranged a "Round the World in 80 days" trip to find out what he could about relevant programming experience. On his return in May 1958 he had some observations. Contrary to his early experience with EDSAC, *computers are assumed to work without error most of the time* which certainly simplified thinking about program design! He was very impressed by new US computer developments (such as Remington-Rand's *LARC* and IBM's *STRETCH*), and he was somewhat surprised that *they use what may be described as conventional techniques*. In particular (Basser, 1958b)

> *I came back with the realisation that there are very few university computing laboratories in the world as well-off as we are as far as equipment is concerned, and, on the whole, our current practices are as efficient as I have seen anywhere.*

And he *had* carried out his brief:

> *Perhaps more important than anything else, I learnt of a number of pit-falls in devising techniques for using magnetic tapes.* (Nucleus, 1958)

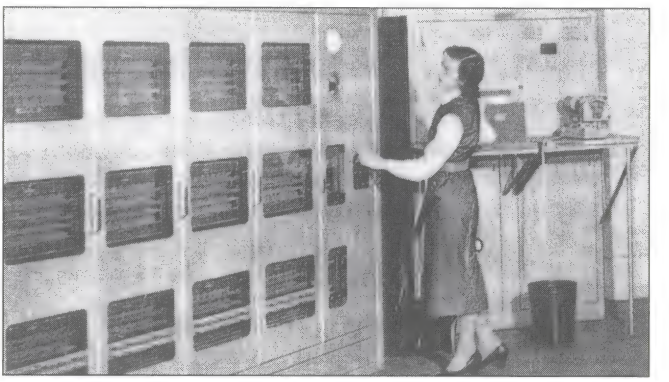


Figure 2: Silliac – the operator is Judy Ogilvie (later the wife of Professor Chris Wallace of Monash University) (Bennett, *et al*, 1994)

John Bennett and Bruce Chartres cobbled together a demonstration program called Z5 for the first Science School (Basser, 1958a). Bennett, Chartres and Elliott(1960) described its origins:

> *Towards the end of 1957, it became necessary to produce a coding scheme suitable for a demonstration of how to code technical problems. This demonstration, lasting 90 minutes, was to follow a lecture of similar length on the general philosophy of computers, and was intended to bring a number of groups of secondary school science masters and mistresses up to the point of at least having some glimmerings of the processes involved in presenting problems to computers. As less than one month was available to complete preparations, considerable use was made of an existing floating point scheme designed by Dr. D. Wheeler while at Illinois in 1953.*

As its popularity increased Z5's classification changed to *utility*, and it took the next available designation A8, progressing to the more flexible A9 in 1958. The changes were a result of experience using it for course work, and many used the scheme regularly for their own work even though

> *The scheme, being interpretive, is about 20 times as slow as normal code for the same work. However, with a machine the speed of SILLIAC, this factor is of little consequence with most problems.*

Bennett *et al* (1960) explained its popularity:

> *All order types required have been defined on a single quarto sheet, and it is perhaps for this reason that no difficulty is experienced from the unfamiliar and apparently arbitrary form of the instructions*

and he wrote later (Bennett, 1998)

> *It was almost user friendly and caught on in a rather embarrassing way. When the KDF9 was introduced, so many programs had been written in it that it was necessary to supply a special compiler to handle them.*

For the same reason a version was produced by the Post Office when they bought their own NCR computer (Bennett, 1990).

In 1959, the University Senate approved an upgrade of the SILLIAC courses to a Diploma in Numerical Analysis and Automatic Computing, and later to the Diploma in Computer Science.

The new year saw a new course: *Simplified Coding*, ie how to use *A9*. John Bennett wrote (Basser, 1959)

> *It was found that three mornings sufficed to bring students with no previous knowledge of automatic computers to a stage of being able to code up simple problems for themselves.*

No wonder it was popular!

While the Basser Technical Reports (1958a, b) mainly examined issues in numerical analysis the fascinating *Some Frontiers in Computer Research* by Bennett (1959) began with a startling survey

> *Automatic digital computers have been with us now for a little over a decade. During this period, we have seen them increase in speed ... accompanied by improved facilities ... Perhaps even more important is the improvement in machine reliability*

> *... Moreover, automatic computers are now widely available. There are more than 3,000 in operation in the Western world ... In Australia, there are nine computers in operation ...*

That was August 1959, and that number was about to increase dramatically. Reliable transistors were just becoming available and integrated circuits were still ten years in the future. But the limitations of the technology did not diminish the enthusiasm of those working with the first generation machines. The possibilities were all around them and progress was rapid. John Bennett stuck his neck out:

> *I would like to quote four predictions made recently ...*
> *(1) Within ten years, a digital computer will be the world's chess champion, unless the rules bar it from competition.*
> *(2) Within ten years, a digital computer will discover and prove an important new mathematical theorem.*
> *(3) Within ten years, a digital computer will write music that will be accepted by the critics as possessing considerable aesthetic value.*
> *(4) Within ten years, most theories in psychology will take the form of computer programs, or of qualitative statements about computer programs.*
> *The writer of this article has little doubt that these predictions will be realised in the time stated.*

Forty years has possibly taught us something about our own complexity. The report went on to mention areas of computing research including: heuristic problem solving, learning , formal logic, musical composition, language translation, pattern recognition, speech synthesis, speech recognition, and hardware topics such as functional redundancy, parallel operation and superconducting logic devices. John Bennett concluded with a reaction to his chosen area of work:

> *I have described briefly some of the "off-beat" fields of research for which automatic computers are being used. So far the surface has hardly been scratched and the exciting thing about the work is that, as with any new tool, practically everything one does has not been tried before.*

The School of Physics received major gifts during 1959 which resulted in a reorganisation and the Basser Computing Laboratory became the Basser Computing Department. Subsequently Bennett became Australia's first Professor of Computing when he was formally installed as Chair of Physics (Electronic Computing).

7. COMPUTING FOR THE MASSES

The First Conference on Automatic Computing and Data Processing in Australia explored many of these *exciting things*. It was held in May 1960 in multiple parallel streams at Sydney University *and* the University of NSW. The title is somewhat curious as there were Australian computer conferences in 1951 and 1957, but this was the first run by the Australian National Committee on Computing and Automatic Control (ANCCAC – the forerunner of the Australian Computer Society) – chaired by John Bennett.

The reports of SILLIAC work started with some automated accounting in Bennett and Whitfeld's (1960) paper which described a system for generating monthly use statistics and user bills – with the details stored on paper tape. They pointed out that:

> *Even a machine with a comparatively small store, which is designed specifically for scientific computations, can be used quite economically for routine data processing operations.*

which had become something of a chore over three years as SILLIAC's use grew.



Figure 3: JMB punching tape (on a 1926 Flexowriter) Millar(1968)

John Bennett (1960) presented some strong advertising to the engineers in *In Lieu of Diagrams and Models*. He argued that while analysis of structures and electrical circuits had been done by "diagrammatic representation" and by models,

> *With the advent of digital computers, a re-orientation of viewpoint is desirable. ... it becomes almost a necessity to think in terms of matrices representing the various entities usually encountered in diagrammatic form ... This prospect may seem startling to someone who has never given it a real trial. However, my own experience – which is shared by many others – is that this approach gives a new insight into the mechanism of solution, and makes it much easier to borrow techniques from other fields where the same underlying mathematics is used.*

The SMHA had arranged to have their own computer (*SNOCOM*) built and they wanted to be ready to use it as soon as possible so they asked for advice. What followed was described by Bennett and Dakin(1960c) in *Computers as an aid in computer design assessment*. Bennett thought they might use a technique he had developed on EDSAC in 1949 and again when Ferranti UK produced their *Mark 1**. Its revised instruction set was *interpreted* by a program run on their old machine. He worked with the SMHA's Milton Chappel to write a SNOCOM interpreter and then to develop its mathematical routine library ready for when the hardware was completed. Bennett (1999) said

> *I gather it worked very well though the two SNOCOM programs with no input or output which were used for tests, when carried out in SILLIAC, took about three times as long as when carried out in SNOCOM itself.*

8. SILLIAC'S RETIREMENT

During the early 1960s computing systems became available with enormously greater capacity than SILLIAC. In addition they were based on new technologies – transistor logic and magnetic core memory – which promised much greater reliability. Harry Messel was instrumental in raising the money to buy an English Electric Leo KDF9. To ease the transition Bennett (Basser, 1964) decided to simulate SILLIAC on the new machine and to provide an A9 Translator. However, it is hard to know if it was an embarrassment or a point of pride that:

> *It has been found that the A9 Translator is 200 times faster than A9 on SILLIAC, and 200 times faster than ALGOL. Also, ALGOL only finds one error per run, whereas A9 offers full diagnostic facilities.*

Not that SILLIAC was fit only for scrap. It had four sophisticated magnetic tape drives, and Basser had the ability to build a high-speed inter-processor link between it and the KDF9. By the time of the *Third Australian Computer Conference* in 1966 there was little front-line computing being done on Basser's old lady, but Bennett, Wallace and Winings(1966) laid out the plans which would become *BasserNet*, one of the first local area networks in the world. The KDF9 – SILLIAC link could be seen as a mainframe – frontend link similar to the IBM 7040 – IBM 1401 pairing that IBM was to donate. They also intended to purchase a Control Data CDC 1700 from an ARGC grant as a second type of front-end to the KDF9, and a PDP8/338 with graphics facilities which could also be linked to the 1401 – all the computers networked together and linked to a dozen user terminals. This worked so well that it was fairly easy to implement a demonstration Unix network a few years later despite the three generations of computer technology embodied in these machines: SILLIAC used valve technology and processed words in parallel, the KDF9 was a germanium transistor solid state machine but did not use integrated circuitry, and the CDC 1700 was partly integrated circuits. However, Bennett and his team had no illusions about SILLIAC's future:

> *In due course, SILLIAC's useful life will come to an end. (We have at present the world's complete stockpile of suitable storage CRTs.)*

No paper on SILLIAC would be complete without mentioning a unique Australian addition to the hardware – the built in beer cooler in the heat exchangers. On May 17, 1968 the last two bottles were ceremoniously extracted and consumed and SILLIAC was closed down. John Bennett spoke about his department's history and pointed out that *over two thousand people made their first contact with computers on SILLIAC* (Basser, 1968).

But Basser and Professor John Bennett went on from strength to strength.

ACKNOWLEDGEMENTS

The people who made it possible to recount the early experiences include not only those mentioned in this paper, but the many with whom subsequent involvement in the practice of IT confirmed the conclusions drawn.

The latter section of this paper is adapted from Deane's *SILLIAC – Vacuum Tube Supercomputer* (in preparation) and much of the earlier section from the personal recollections of de Ferranti.

GENERAL REFERENCES

ANON (1978): Twenty-one years of computing. in The University of Sydney Gazette, May.
BENNETT, J. (1994): Computer Science at the University of Sydney 1953-1986. BasserNet 3(6), December.
PEARCEY, T. (1988): A History of Australian Computing. Chisholm Institute of Technology

SPECIFIC REFERENCES

AUSTRALIAN GOVERNMENT (1968): Employment and Technology. Dept Labour and National Service Report Series.
BASSER (1956): SILLIAC Programming Manual.
BASSER (1958a): Quarterly Operational Report. 7, January-March.
BASSER (1958b): Quarterly Operational Report. 8, April-June.
BASSER (1959): Quarterly Operational Report 11, January-March.
BASSER (1964): News Sheet 9, 29 August.
BASSER (1968): News Sheet 20, 21 June.
BENNETT, J. (1956a): letter To Professor Messel. 5 March.

BENNETT, J. (1956b): record of Visit to Snowy Mountains Hydro-Electric Authority. 6-9 March.
BENNETT, J. (1956c): letters to Institution of Engineers, & to Institute of Accountants. 21 March.
BENNETT, J. (1956d): letter to Dr Ralph Meagher. 17 April.
BENNETT, J. (1956e): letter to Snowy Mountains Hydro-electric Authority. 29 June.
BENNETT, J. and SWIRE, B. (1957): The SILLIAC. *Proc. Data Processing & Automatic Computing Machines, Part I.* Weapons Research Establishment, Salisbury, South Australia.p.103.
BENNETT, J. (1957a): The use of continuants in practical numerical analysis. *Proc. Data Processing & Automatic Computing Machines, Part I.* Weapons Research Establishment, Salisbury, South Australia.
BENNETT, J. (1957b): A Year's Operating Experience with SILLIAC. Basser Technical Report 1, September.
BENNETT, J. (1958a): Proposal for a regression analysis program for SILLIAC. Basser Technical Report 2, January.
BENNETT, J. (1958b): Proposals for solving large sets of linear simultaneous equations. Basser Technical Report 3, March.
BENNETT, J. (1959): Some frontiers in computer research. Basser Technical Report 6, August.
BENNETT, J., CHARTRES, B. and ELLIOTT, JOY (1960): Simplified coding - a pedagogic experiment. *Proc. of First Conference on Automatic Computing and Data Processing in Australia.* ANCCAC.
BENNETT, J. (1960): In lieu of diagrams and models. *Proc. of First Conference on Automatic Computing and Data Processing in Australia.* ANCCAC.
BENNETT, J. and DAKIN, R. (1960): Computers as an aid in computer design assessment. *Proc. of First Conference on Automatic Computing and Data Processing in Australia.* ANCCAC.
BENNETT, J. and WHITFIELD, R. (1960): Data processing with paper tape, an experiment. *Proc. of First Conference on Automatic Computing and Data Processing in Australia.* ANCCAC.
BENNETT, J., WALLACE, C. and WININGS, J. (1966): Software and hardware contributions to efficient operating on a limited budget. *Proc. of the Third Australian Computer Conference, Canberra*: 127-129.
BENNETT, J. (1990): Early computer days in Britain and Australia - some autobiographical snippets. *Annals of the History of Computing* 12(4): 281-285.
BENNETT, J., BROOMHAM, R., MURTON, P., PEARCEY, T. AND RUTLEDGE, R. (1994): Computing in Australia - the development of a profession. Hale and Iremonger.
BENNETT, J. (1998): Autobiographical Snippets. In *In the beginning – personal recollections of software pioneers.* GLASS, R.(ed). IEEE Computer Society Press: 212-237.
BENNETT, J. (1999): comment during a talk given by Kevin Rosolen at the Powerhouse Museum. 16 August.
BENNETT, J. (2001): email to author. March.
DE FERRANTI, B. (1956): The Electronic Computer "SILLIAC" and the Adolph Basser Computing Laboratory. Nuclear Research Foundation, University of Sydney. September.
MILLAR, D. (1987): The Messel Era - The story of the School of Physics and its Science Foundation within the University of Sydney 1952 - 1987. Pergamon.
NATIONAL ARCHIVES (2001): Series A1200. Caption Number L21782.
http://www.naa.gov.au/scripts/aaserve.pl?doc=/aa/aacaps/A1200/82/L21782.html
NUCLEUS (1955): Nuclear Research Foundation, University of Sydney 1(8)
NUCLEUS (1958): Nuclear Research Foundation, University of Sydney 4(2)

BIOGRAPHICAL NOTES

Barry Zani de Ferranti is a pioneer in Australian computing: involved in the earliest research on computers at the University of Sydney and CSIRO, then in building SILLIAC, he later introduced IBM's first computers and then established FERRANTI LTD's first computing sales office in Australia. After periods in industry, he founded Barry Ferranti Associates Pty. Limited and Perspective Management Services Pty. Ltd. (AUERBACH Publishers, etc., agency), where he has carried out a wide variety of computer consulting projects in industry, education and government. He is also well known as a lecturer and author of numerous articles and reviews, and computing textbook contributor. More recently he has focussed on Internet software and applications.

John Deane has a B.A. in mathematics/computing from Macquarie University. Following three years commercial programming, he joined CSIRO Radiophysics in 1974. He has wide experience in real-time programming including development of control systems for the Parkes 64m telescope and Australia Telescope. Networking experience gained there and during a period as computing systems manager brought him into a wireless networking program where he contributed to two patents. Over this period he has written two introductory programming books (with Judy Deane),

'A Picture History of Radiophysics' and three booklets (on CSIRAC, the first modern computer and an industrial controller) for the Australian Computer Museum Society. He currently edits their newsletter and is working on a book about Sydney University's first computer 'SILLIAC'.

A Network Flow Method for Determining the Gradient of Roads

A. Land
London School of Economics

The paper describes a conversion of the road "cut and fill" design problem for solution by a network algorithm.

Keywords: cut and fill design, network algorithm
CR Categories: G.1.6 – Constrained optimisation, G.4 – Algorithm design and analysis, J.2 – Computer applications in engineering

1. INTRODUCTION

There is a class of mathematical programming models which is so very much simpler and faster to solve than even the general linear programming problem that it sometimes warrants some distortion of the real-world problem to make it fit. This class is the totally unimodular linear programming problems, of which the most familiar examples are the so-called Transportation problem and the Assignment problem. This paper is to draw attention to another example, which arises in the field of road design – in particular for a network of residential roads.

Once the plan of the road network has been determined, a second design stage is to determine the vertical profiles of the roads. Since there are constraints on the profiles, and costs associated with cutting and filling, it is not surprising that a mathematical programming model should be applicable. Under the following assumptions it is possible to have a unimodular formulation.

1.1 Assumptions

1. The plan of the road system has been decided and only the vertical profiles remain to be determined.
2. Only the gradients of the roads are subject to constraint, not curvatures (second derivative of vertical position). Other constraints take the forms of limitations on height differences, or specified heights, only.
3. An adequate representation of the ground contours can be given by a limited grid of points in the plane.
4. Cost of construction can be adequately approximated by a linear function (or piecewise linear convex function) of the height of the road above or below the ground surface at each grid point.
5. The gradient constraints can be adequately represented by maximum absolute differences of the road between (normally, but not necessarily, geographically adjacent) grid points.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

1.2 Data

n is the number of grid points.

m is the number of constraints on height differences between pairs of grid points.

h_j is the height of the ground at grid point j, measured (in metres) from an arbitrary zero, e.g. sea-level.

g_{ij} is the maximum allowed increase in road height (above sea-level) between grid points i and j. (g_{ij} need not be equal to g_{ji} and may depend on their distances apart, of course.)

a_j is the cost per metre of height of fill (embankment) for the area associated with grid point j.

b_j is the cost per metre of cut for the area associated with grid point j.

The costs, per metre, a_j and b_j are initially assumed constant, but piece-wise linear convex functions could easily be included.

Figure 1 illustrates a proposed network of roads (shown as straight lines) and the contours of the ground level. The n grid points are represented by the dots on the network.

1.3 Variables

z_j is the height above sea-level of the road at grid point j.

f_j is the height of the fill at grid point j.

c_j is the depth of cut at grid point j.

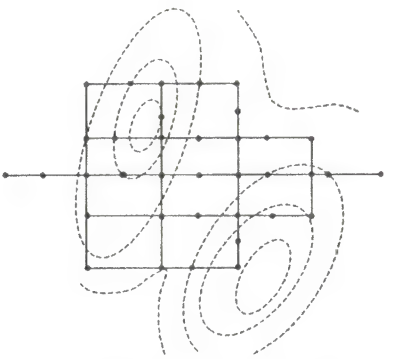


Figure 1: Roads and Contours

2. FORMULATION

The problem is then to

$$\text{Minimise} \quad \sum_{j=1}^{n} a_j f_j + \sum_{j=1}^{n} b_j c_j$$

$$\text{subject to} \quad \left.\begin{array}{l} z_i - z_j \le g_{ij} \\ z_j - z_i \le g_{ji} \end{array}\right\} \text{(1) for all } m \text{ gradient constraints between points } i \text{ and } j$$

$$\left.\begin{array}{l} f_j \ge z_j - h_j \\ f_j \ge 0 \end{array}\right\} \text{(2) definition of height of embankment}$$

$$\left.\begin{array}{l} c_j \ge h_j - z_j \\ c_j \ge 0 \end{array}\right\} \text{(3) definition of depth of cutting}$$

This linear program can be solved by a minimum cost flow diagram on n+1 nodes and $2n$+m arcs. The details of this transformation will be found in the appendix. Each linear segment of a convex cost function at a grid point will add one more extra arc to the network, but will not affect

the nature of the calculation. A standard method such as Kennington and Helgason (1980) or NETFLOW (neos) can be used to give an extremely rapid solution to this type of problem, although no doubt a more specific algorithm could be written, for instance to handle a quadratic cost function.

The main point, however, is that the solution procedure should be sufficiently trivial to enable this algorithm to be used as a subroutine and solved repeatedly as part of an iterative process. For instance, if the costs of cutting and embankment are not known constants at each grid point, but are dependent on the costs of transport, of "cut and fill", between parts of the road, then a_j and b_j could be the marginal costs of filling and cutting derived in the major algorithm, and this flow algorithm could be used repeatedly as part of a decomposition of the whole problem.

Assumption 5 may not be valid if there are minimum rather than maximum gradient conditions on a section of the road. There is no way to build minimum gradient conditions into the model, but so long as these conditions were only violated in minor instances in the first run of the algorithm, inequality conditions could be replaced by equalities imposing the required minimum gradients, in a second run. Curvature restrictions (assumed away by assumption 2) could be approximated by imposing a finer grid at points identified by a first run as critical, with more restrictive gradient conditions.

The solution to this model would most realistically be regarded as providing a finite set of points to which cubic splines could be fitted. The resulting cost would not be identical but it seems likely that the major part of the determination of a satisfactory and efficient set of road profiles would be achieved by linear algorithms.

Finally, if is sufficiently fast and economical to solve for some hundreds of grid points, it may be possible to iterate by optimising the cost for each plan of the road system and then modifying the plan in the light of the consequent costs of cut and fill.

REFERENCES

KENNINGTON, J.L. and HELGASON, R.V. (1980): Algorithms for network programming John Wiley & Sons, NewYork.
NEOS: http://www-neos.mcs.anl.gov

APPENDIX

Associate m dual variables, u_{ij}, with constraints (1), n dual variables, v, with constraints (2), and n dual variables, w, with constraints (3), and then the dual problem is:

$$\text{Minimise} \quad \sum^{m} g_{ij}u_{ij} + \sum_{j=1}^{n} h_j v_j - \sum_{j=1}^{n} h_j w_j$$

$$\text{subject to} \quad \sum_j u_{ij} - \sum_j u_{ji} + v_i - w_i = 0 \quad \text{for all } i, \text{ and } i,j \text{ a gradient restricted pair (flow conservation constraints)}$$

$$\left.\begin{array}{l} v_i \le a_i \\ w_i \le b_i \end{array}\right\} \text{ upper bounds on } v_i \text{ and } w_i$$

$$u \ge 0,\ v \ge 0,\ w \ge 0.$$

The variables u_{ij} have only two non-zero elements, +1 and -1 , in their columns of the coefficient matrix, and therefore can be represented as directed arcs between nodes i and j.

Variables v_i and w_i have only one non-zero element, +1, and -1, respectively. The addition of an (n+1)th zero net balance vertex as origin for the arcs $v_{n+1,\,i}$ and destination for the arcs $w_{i,n+1}$ completes the transformation into a problem of minimising "cost" (with some negative costs) subject to zero net flow at n+1 vertices, with m unbounded non-negative arcs and 2n bounded arcs connecting each of the grid points to the dummy, (n+1)th, vertex.

The solution to the road height problem is read from the dual solution of this flow problem, after subtracting the dual value of the dummy vertex from all the other dual values.

Piecewise linear convex functions instead of a_j and b_j can be handled by the addition of extra bounded arcs. Similarly, simple restrictions on the gradients could be replaced by increasingly costly relaxations of the constraints, up to an absolute maximum gradient.

BIOGRAPHICAL NOTES

Ailsa Land has a BSc and PhD from London School of Economics, where she is now an Emeritus Professor of the Operational Research Department, which she formerly headed. Her research interests include many aspects of optimisation and the analysis and formulation of operational problems and the development of appropriate solution methods. She was a frequent visitor to the Basser Department of Computer Science.

On the Complete Pivoting Conjecture for Hadamard Matrices of Small Orders

C. Koukouvinos

Department of Mathematics,
National Technical University of Athens, Zografou 15773, Athens, Greece,
ckoukouv@math.ntua.gr

E. Lappas and M. Mitrouli

Department of Mathematics, University of Athens, Panepistemiopolis 15784, Athens, Greece.

Jennifer Seberry

School of Information Technology and Computer Science, University of Wollongong, Wollongong, NSW, 2522, Australia.

In this paper we study explicitly the pivot structure of Hadamard matrices of small orders 16,20 and 32. An algorithm computing the (n–j) × (n–j) minors of Hadamard matrices is presented and its implementation for n = 12 is described. Analytical tables summarising the pivot patterns attained are given.

Keywords: Gaussian elimination, pivot size, complete pivoting, minors, Hadamard matrices.

AMS Subject Classification: 65F05, 65G05, 20B20.

CR Categories: F.2.1 – Computations on matrices, G.1.3 – Numerical Linear Algebra.

1. INTRODUCTION

Let A be an $n \times n$ real matrix, and let $\underline{b}$ be a real n-vector. In his fundamental work on backward error analysis Wilkinson(1963) proved that when the linear system $A \cdot \underline{x} = \underline{b}$ is solved in floating point arithmetic by Gaussian elimination (GE) with either partial or complete pivoting the computed solution $\hat{\underline{x}}$ satisfies

$$(A+E) \cdot \hat{\underline{x}} = \underline{b}$$

where the norm of the perturbation matrix E can be bounded from above as follows:

$$\|E\|_\infty \leq g(n,A) \cdot f(n) \cdot u \, \|A\|_\infty \tag{1}$$

where u is the unit roundoff, $f(n)$ is a cubic polynomial of n, and $g(n,A)$ is the growth factor defined by

$$g(n, A) = \frac{\max_{i,j,k} |a_{ij}^{(k)}|}{|a_{11}^{(0)}|}$$

where $a_{ij}^{(k)}$, $k = 1,2,\ldots, n-1$ denotes the (i,j) th element that occurs at the k-th step of elimination. The elements $a_{ii}^{(n-1)}$ are called pivots.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: 14 December 2001
Associate Editor: Jenny Edwards

Thus the growth factor is closely related to the stability of GE since the smaller its value the sharper the bound attained in (1) and consequently the more stable the method becomes.

We say that a matrix A is completely pivoted (CP) if the rows and columns have been permuted so that Gaussian elimination with no pivoting satisfies the requirements for complete pivoting.

Let $g(n,A)$ denote the growth associated with Gaussian elimination on a CP $n \times n$ matrix A and $g(n) = \sup \{g(n,A)\}$. The problem of determining $g(n)$ for various values of n is called the *growth problem.*

The determination of $g(n)$ remains a challenging problem.Wilkinson (1963; 1988) noted that there were no known examples of matrices for which $g(n) > n$. In 1968, Cryer conjectured that "$g(n,A) \leq n$, with equality if and only if A is a Hadamard matrix". This was proved to be "untrue" in Gould (1991).

A Hadamard matrix H of order n is an $n \times n$ matrix with elements ± 1 and $HH^T = nI$.

Since Wilkinson's initial conjecture seems to be connected with Hadamard matrices the following, more refined, conjecture was posed (Cryer,1968; Day and Peterson,1988; Mitrouli and Koukouvinos, 1996):

Conjecture (The growth conjecture for Hadamard matrices)
Let A be an $n \times n$ CP Hadamard matrix. Reduce A by GE. Then

(i) $g(n, A) = n$.

(ii) The four last pivots are equal to $\frac{n}{2}$ or $\frac{n}{4}, \frac{n}{2}, \frac{n}{2}, n$.

(iii) The fifth last pivot can take the values $\frac{n}{3}$ or $\frac{n}{2}$.

(iv) The sixth last pivot can take the values $\frac{n}{4}$, $\frac{n}{10/3}$, or $\frac{n}{8/3}$.

(v) Every pivot before the last has magnitude at most $\frac{n}{2}$.

(vi) The first six pivots are equal to $1, 2, 2, 4$, 2 or 3, $\frac{10}{3}$ or $\frac{8}{3}$ or 4.

The equality in (i) above has been proved for a certain class of $n \times n$ Hadamard matrices (Day and Peterson, 1988). Cryer (1968) has shown (ii) for the pivots $\frac{n}{2}, \frac{n}{2}$ and n. Day and Peterson (1988) have shown that the values $\frac{n}{2}$ or $\frac{n}{4}$ appear in the fourth last pivot when Gaussian elimination, not necessarily with complete pivoting, is applied to a Hadamard matrix. They posed the conjecture that when Gaussian elimination with complete pivoting is applied to a Hadamard matrix the value of $\frac{n}{2}$ is impossible. In Edelman and Friedman (1998) a counter example of a Hadamard matrix of order 16 which has fourth last pivot $\frac{n}{2}$ is given. In Koukouvinos, Mitrouli and Seberry (to appear), ten CP Hadamard matrices of order 16 are given having fourth last pivot 8, as well as a CP Hadamard matrix of order 32 with fourth last pivot 16. In each case this distinguished case with the fourth pivot $\frac{n}{2}$ arose in the equivalence class containing the Sylvester Hadamard matrix. Moreover, in Koukouvinos *et al* (to appear), the first infinite family of Hadamard matrices of order n with fourth last pivot $\frac{n}{2}$ is discovered. The values in (vi) are proved in Day and Peterson (1988) for the first five values, 1, 2, 2, 4, 2 or 3, and experimental evidence in Mitrouli and Koukouvinos (1996) and this paper strongly supports the next values.

Wilkinson's initial conjecture seems to be connected with Hadamard matrices. Interesting results in the size of pivots appear when GE is applied to CP skew-Hadamard and weighing matrices of order n and weight n–1. In these matrices, the growth is also large, and experimentally,

we have been led to believe it equals n–1 and special structure appears for the first few and last few pivots (Koukouvinos *et al*, 2000).

Notation 1. Write A for a matrix of order n, which is completely pivoted (CP). Write $A(j)$ for the absolute value of the determinant of the $j \times j$ principal submatrix in the upper lefthand corner of the matrix A and $A[j]$ for the absolute value of the determinant of the $(n–j) \times (n–j)$ principal submatrix in the bottom righthand corner of the matrix A. The magnitude of the pivots appearing after the application of GE operations is given by

$$p_j = A(j)/A(j–1). \tag{2}$$

Notation 2. We use – for –1 in matrices in this paper. Also, when we are saying the determinants of a matrix we mean the absolute values of the determinants.

2. MINORS OF HADAMARD MATRICES

Hadamard matrices of order n have determinant $n^{\frac{n}{2}}$. Sharpe (1907) observed that the determinants of all the $(n–1) \times (n–1)$ minors of a Hadamard matrix of order n are zero or $n^{\frac{n}{2}-1}$, and that the determinants of all the $(n–2) \times (n–2)$ minors are zero or $2n^{\frac{n}{2}-2}$, and the determinants of all the $(n–3) \times (n–3)$ minors are zero or $4n^{\frac{n}{2}-3}$. We note that the maximum determinant corresponds to having the submatrix

$$\begin{bmatrix} 1 & 1 \\ 1 & - \end{bmatrix} \text{ or } \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & - \\ 1 & - & 1 \end{bmatrix}$$

in the upper lefthand corner of the Hadamard matrix for n–2 and n–3 respectively.

Theorem 1. *[Koukouvinos* et al, *2001] The determinants of the (n–3) × (n–3) minors of a CP Hadamard matrix of order n are* $4n^{\frac{n}{2}-3}$.

Theorem 2. *[Koukouvinos* et al, *2001] The determinants of the (n–4) × (n–4) minors of a CP Hadamard matrix of order n are* $8n^{\frac{n}{2}-4}$ *or* $16n^{\frac{n}{2}-4}$.

A direct consequence of the above theorems and of relation (2) is the following:

Theorem 3. *When Gaussian Elimination is applied on a CP Hadamard matrix H of order n the last four pivots are n,* $\frac{n}{2}, \frac{n}{2}, \frac{n}{2}$ *or* $\frac{n}{4}$.

We now outline the method to evaluate the $(n–5),(n–6),\ldots,(n–j)$ minors. The $(n-j) \times (n–j)$ minors are denoted by M_{n-j}.

Lemma 1. *(The Distribution Lemma) [Koukouvinos* et al, *2001] Let H be any Hadamard matrix, of order n > 2. Then for every triple of rows of H there are precisely* $\frac{n}{4}$ *columns which are*

(a) $(1,1,1)^T$ *or* $(–,–,–)^T$

(b) $(1,1,–)^T$ *or* $(–,–,1)^T$

(c) $(1,–,1)^T$ *or* $(–,1,–)^T$

(d) $(1,–,–)^T$ *or* $(–,1,1)^T$

If we are considering the $(n-j) \times (n-j)$ minors, then the first j rows, ignoring the upper lefthand $j \times j$ matrix, have 2^{j-1} potentially different first j elements in each column. Let $\underline{x}_{\beta+1}^{T}$ be the vectors containing the binary representation of each integer $\beta+2^{j-1}$ for $\beta=0,...,2^{j-1}-1$. Replace all zero entries of $\underline{x}_{\beta+1}^{T}$ by -1 and define the $j \times 1$ vectors

$$\underline{u}_k = \underline{x}_{2^{j-1}-k+1},\ k=1,...,2^{j-1}$$

Let u_k indicate the number of columns beginning with the vectors $\underline{u}_k$, $k=1,...,2^{j-1}$. We note

$$\sum_{i=1}^{2^{j-1}} u_i = n - j. \tag{3}$$

Then it holds (Koukouvinos *et al*, 2001) that

$$M_{n-j} = n^{n-2^{j-1}-j} det D \tag{4}$$

where D is the following $2^{j-1} \times 2^{j-1}$ matrix.

$$D = \begin{bmatrix} n-ju_1 & u_2m_{12} & u_3m_{13} & \cdots & u_zm_{1z} \\ u_1m_{21} & n-ju_2 & u_3m_{23} & \cdots & u_zm_{2z} \\ \vdots & \vdots & \vdots & & \vdots \\ u_1m_{z1} & u_2m_{z2} & u_3m_{z2} & \cdots & n-ju_z \end{bmatrix}$$

where $(m_{ik}) = (-\underline{u}_i \cdot \underline{u}_k)$, with $\cdot$ the inner product.
We note if $n = 4t$, the orthogonality of the rows of the Hadamard matrix gives

$$t-j \le u_1+u_2+\ldots+u_{2j-3} \le t, \quad t-j \le u_{2j-3+1}+\ldots+u_{2j-2} \le t$$

$$t-j \le u_{2j-2+1}+\ldots+u_{2j-2+2j-3} \le t, \quad t-j \le u_{2j-2+2j-3+1}+\ldots+u_{2j-1} \le t$$

Each of these equations can be rewritten so the constraints become

$$\begin{aligned} 0 \le t-u_1-u_2-\ldots-u_{2j-3} \le j, \quad 0 \le t-u_{2j-3+1}-\ldots-u_{2j-2} \le j \\ 0 \le t-u_{2j-2+1}-\ldots-u_{2j-2+2j-3} \le j, \quad 0 \le t-u_{2j-2+2j-3+1}-\ldots-u_{2j-1} \le j \end{aligned} \tag{5}$$

The Algorithm for $(n-j) \times (n-j)$ minors of an $n \times n$ Hadamard matrix H.

Step 1: Generate all ± 1 matrices M, of order j.

Step 2: Form the general matrix, $N = [M\ U_j]$, of size $j \times n$ for the first j rows of an $n \times n$ Hadamard matrix H, where

$$U_j = \begin{array}{ccccc} \overbrace{1\ldots1}^{u_1} & \overbrace{1\ldots1}^{u_2} & & \overbrace{1\ldots1}^{u_{2j-1-1}} & \overbrace{1\ldots1}^{u_{2j-1}} \\ 1\ldots1 & 1\ldots1 & \ldots & 1\ldots1 & 1\ldots1 \\ 1\ldots1 & 1\ldots1 & \ldots & -\ldots- & -\ldots- \\ . & . & \ldots & . & . \\ . & . & \ldots & . & . \\ 1\ldots1 & 1\ldots1 & \ldots & 1\ldots1 & -\ldots- \\ 1\ldots1 & -\ldots- & \ldots & -\ldots- & -\ldots- \end{array}$$

Step 3: For each M consider all $\binom{j}{3}$ subsets of three rows of N and use the Distribution Lemma with $\sum_{i=1}^{2^{j-1}} u_i = n - j$ to form 4 equations in the variables $u_1,...,u_{2^{j-1}}$ for each subset.

Step 4: For each M search for all feasible solutions to the different equations generated at Step 3.

Step 5: For each M and for each feasible solution found in Step 4 use the matrix D to find all possible values of the $(n–j) \times (n–j)$ minors.

Implementation of the algorithm

Let $n = 12$ and $j = 5$. We demonstrate analytically the implementation of the algorithm.

Step 1-Step 2:

For each 5×5 matrix M with elements ± 1 and first row and column all 1 we form the matrix $N = [M, U_5]$. There are 2^{16} different matrices M. The first five rows of matrix N are:

	v_1				v_2				v_3				v_4			
	z_1		z_2		z_3		z_4		z_5		z_6		z_7		z_8	
	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}	u_{15}	u_{16}
M	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	–	–	–	–	–	–	–	–
	1	1	1	1	–	–	–	–	1	1	1	1	–	–	–	–
	1	1	–	–	1	1	–	–	1	1	–	–	1	1	–	–
	1	–	1	–	1	–	1	–	1	–	1	–	1	–	1	–

Recalling that $u_1,u_2,...,u_{16}$ are the number of columns of the kind indicated. From the order we have

$$\sum_{i=1}^{16} u_i = 7. \tag{6}$$

We first see by the Distribution Lemma that

$$0 \le u_i \le 3. \tag{7}$$

Step 3 – Step 4:

In what follows, m_{ij} denotes the inner product of rows i and j of the matrix N.

(I) For every triple of rows (i,k,l) of N let v_1,v_2,v_3,v_4 be the number of columns starting correspondingly with:

(a) $(1,1,1)^T$ or $(–,–,–)^T$
(b) $(1,1,–)^T$ or $(–,–,1)^T$
(c) $(1,–,1)^T$ or $(–,1,–)^T$
(d) $(1,–,–)^T$ or $(–,1,1)^T$

By the orthogonality of these three rows and the order we form the equations:

$$\begin{aligned} v_1+v_2-v_3-v_4 &= -m_{ik} \\ v_1-v_2+v_3-v_4 &= -m_{il} \\ v_1-v_2-v_3+v_4 &= -m_{kl} \\ v_1+v_2+v_3+v_4 &= 7 \end{aligned}$$

This system can be solved uniquely

$$\begin{aligned} v_1 &= \frac{1}{4}(7 - m_{ik} - m_{il} - m_{kl}) \\ v_2 &= \frac{1}{4}(7 - m_{ik} + m_{il} + m_{kl}) \\ v_3 &= \frac{1}{4}(7 - m_{ik} - m_{il} + m_{kl}) \\ v_4 &= \frac{1}{4}(7 + m_{ik} + m_{il} - m_{kl}) \end{aligned} \tag{8}$$

The distribution lemma gives that $0 \le v_i \le 3$, i =1,2,3,4 and v_i must be integer. Thus for each M and for all possible triplets (there are $\binom{5}{3}$ =10 triplets) of rows if there is at least one triplet and at least one m: $v_m \ge 4$ or $v_m < 0$ or v_m not integer, then we reject this matrix M. Finally the number of possible matrices M is reduced to 40860.

We call the vector $ip=(m_{12}, m_{13}, m_{14}, m_{15}, m_{23}, ..., m_{25}, ..., m_{45})$,

for a given M the *inner product (IP) profile vector*. We call two matrices inner product equivalent if they have the same (IP). Hereinafter we can work with the (IP) instead of M. We found 1975 inequivalent (IP) profile vectors for the 40860 M found previously.

For each (IP) and for the first three lines we calculate v_4.

(II) For the first four lines of N we notice that there are eight different columns $z_1,z_2,...,z_8$. Using the orthogonality and the order and solving the system we have:

$$\begin{aligned} z_1 &= 1/4 \cdot (7 - m_{23} - m_{24} - m_{34}) - z_8 \\ z_2 &= 1/4 \cdot (m_{34} - m_{13} + m_{24} - m_{12}) + z_8 \\ z_3 &= 1/4 \cdot (+m_{34} - m_{14} + m_{23} - m_{12}) + z_8 \\ z_4 &= -1/4 \cdot (7 - m_{34} + m_{13} + m_{14}) - z_8 \\ z_5 &= 1/4 \cdot (m_{24} - m_{14} + m_{23} - m_{13}) + z_8 \\ z_6 &= 1/4 \cdot (7 - m_{24} + m_{12} + m_{14}) - z_8 \\ z_7 &= 1/4 \cdot (7 - m_{23} + m_{12} + m_{13}) - z_8 \end{aligned} \tag{9}$$

For a given *(IP)* and its v_4 as found from (8) the possible values of the parameter z_8 are from 0 to v_4 because $z_7 + z_8 = v_4$ and thus $z_8 = v_4 - z_7 \le v_4$ (z_7 is always a non negative number). For all the possible values of z_8 we compute the z_i checking also if $0 \le z_i \le 3$ and z_i integer. From all the possible solutions we need for the next step the vector containing the (IP) and the coordinates z_4, z_6, z_7, z_8 of the solutions denoted by [IP, z_4, z_6, z_7, z_8].

(III) By the orthogonality of the five rows of N, the order and by solving the system of the equations produced we get:

$$\begin{aligned} u_1 &= 1/2 \cdot 7 - 1/4 \cdot (m_{45} + m_{25} + m_{35} - m_{15} - m_{12} + m_{23} - m_{13} \\ &\quad + m_{24} + m_{34} - m_{14}) - u_8 - u_{12} - u_{14} - u_{15} - 3 \cdot u_{16} \\ u_2 &= -1/4 \cdot (7 - m_{45} - m_{25} - m_{35} + m_{15} + m_{12} + m_{13} + m_{14}) \\ &\quad + u_8 + u_{12} + u_{14} + 2 \cdot u_{16} \end{aligned}$$

$$
\begin{aligned}
u_3 &= -1/4 \cdot (7 - m_{45} + m_{15} + m_{12} + m_{13} - m_{24} - m_{34} + m_{14}) \\
&\quad + u_8 + u_{12} + u_{15} + 2 \cdot u_{16} \\
u_4 &= 1/4 \cdot (7 - m_{45} + m_{15} + m_{14}) - u_8 - u_{12} - u_{16} \\
u_5 &= -1/4 \cdot (7 - m_{35} + m_{15} + m_{12} - m_{23} + m_{13} - m_{34} + m_{14}) \\
&\quad + u_8 + u_{14} + u_{15} + 2 \cdot u_{16} \\
u_6 &= 1/4 \cdot (7 - m_{35} + m_{15} + m_{13}) - u_8 - u_{14} - u_{16} \\
u_7 &= 1/4 \cdot (7 - m_{34} + m_{13} + m_{14}) - u_8 - u_{15} - u_{16} \\
u_9 &= -1/4 \cdot (7 - m_{25} + m_{15} + m_{12} + m_{13} - m_{23} - m_{24} + m_{14}) \\
&\quad + u_{12} + u_{14} + u_{15} + 2 \cdot u_{16} \\
u_{10} &= 1/4 \cdot (7 + m_{15} - m_{25} + m_{12}) - u_{12} - u_{14} - u_{15} \\
u_{11} &= 1/4 \cdot (7 + m_{14} - m_{24} + m_{12}) - u_{12} - u_{15} - u_{16} \\
u_{13} &= 1/4 \cdot (7 + m_{13} - m_{23} + m_{12}) - u_{14} - u_{15} - u_{16}
\end{aligned}
\tag{10}
$$

For each [IP, z_4, z_6, z_7, z_8] found from (9) we specify the possible values of the parameters u_8, u_{12}, u_{14}, u_{16}. These are $0 \le u_8 \le z_4$, $0 \le u_{12} \le z_6$, $0 \le u_{14} \le z_7$ and $0 \le u_{16} \le z_8$. u_{15} is not a "real" parameter because $u_{15} + u_{16} = z_8$ and thus $u_{15} = z_8 - u_{16}$.

Step 5: For each [u_1, u_2, ..., u_{16}] and with the use of matrix D we find the possible values of the (12–5) × (12–5) minors. Thus the 7×7 minors can take the values {0,192,384,576}.

Expanding the above algorithm for j=6 we find as possible values for the 6×6 minors the following: {0,32,64,96,128,160}

Testing the above algorithm up to the (n–6) case we can summarise our results as follows:

order	Values of Minors
n	$n^{\frac{n}{2}}$
n-1	$0,\ n^{\frac{n}{2}-1}$
n-2	$0,\ 2n^{\frac{n}{2}-2}$
n-3	$0,\ 4n^{\frac{n}{2}-3}$
n-4	$0,\ 8n^{\frac{n}{2}-4},\ 16n^{\frac{n}{2}-4}$
n-5	$0,\ 16n^{\frac{n}{2}-5},\ 32n^{\frac{n}{2}-5},\ 48n^{\frac{n}{2}-5}$
n-6	$0,\ 32n^{\frac{n}{2}-6},\ 64n^{\frac{n}{2}-6},\ 96n^{\frac{n}{2}-6},\ 128n^{\frac{n}{2}-6},\ 160n^{\frac{n}{2}-6}$

Table 1

Remark 1 We see that as the value of n–j decreases the range of values of the corresponding minors M_{n-j} increases.

3. PIVOT STRUCTURE OF THE 16 X 16 HADAMARD MATRICES

Hall (1961) and Seberry Wallis (1972) proved that there are five equivalence classes for Hadamard matrices of order 16 and gave examples of each.

In our subsequent experiments we took 40000 cases from each of the five equivalence classes and applied GECP to each. We found 9 different pivot patterns for class I. For class II we found 18 different pivot patterns. For class III we found 21 different pivot patterns.

Since classes IV and V are one another's transpose they are identical for the purpose of GECP (Day and Peterson, 1988). We found classes IV and V gave 12 different pivot patterns but each of these patterns had already occurred in classes I, II or III. The following tables summarise the different pivot structures attained for each class and highlight the difference between the classes. We note that the fourth last pivot 8 only occurs in class I which is the equivalence class of Sylvester Hadamard matrices.

	growth	Class I- Pivot Pattern
1	16	$(1,2,2,4,2,4,4,\ 8,\ 2,4,\ 4,\ 8,4,8,8,16)$
2	16	$(1,2,2,4,2,4,4,\frac{16}{8/3},\frac{8}{3},4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
3	16	$(1,2,2,4,2,4,4,\frac{16}{8/3},\frac{8}{3},4,\ 4,\ 8,4,8,8,16)$
4	16	$(1,2,2,4,3,\frac{8}{3},4,\frac{16}{8/3},\frac{8}{3},4,\ 4,\ 8,4,8,8,16)$
5	16	$(1,2,2,4,3,\frac{8}{3},4,\frac{16}{8/3},\frac{8}{3},4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
6	16	$(1,2,2,4,3,\frac{8}{3},2,\ 4,\ 4,8,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
7	16	$(1,2,2,4,3,\frac{8}{3},2,\ 4,\ 4,8,\ 4,\ 8,4,8,8,16)$
8	16	$(1,2,2,4,3,\frac{8}{3},2,\ 4,\ 4,4,\ 8,\ 8,4,8,8,16)$
9	16	$(1,2,2,4,3,\frac{8}{3},2,\ 4,\ 4,4,\ 4,\ 8,8,8,8,16)$

Table 2

	growth	Class II- Pivot Pattern
1	16	$(1,2,2,4,2,4,4,\frac{16}{16/5},\frac{16}{5},\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
2	16	$(1,2,2,4,2,4,4,\frac{16}{16/5},\frac{16}{5},\ 4\ ,\ \frac{16}{8/3},\frac{16}{3},4,8,8,16)$
3	16	$(1,2,2,4,2,4,4,\ 4\ ,\ 4\ ,\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
4	16	$(1,2,2,4,2,4,4,\ 4,\ 4,\ 4,\ \frac{16}{8/3},\ \frac{16}{3},4,8,8,16)$
5	16	$(1,2,2,4,2,4,4,\ 4,\ 4,\ 4,\ \ 4,\ 8,\ 4,8,8,16)$
6	16	$(1,2,2,4,2,4,4,\frac{16}{16/5},\frac{16}{5},\ 4,\ \ 4,\ 8,4,8,8,16)$
7	16	$(1,2,2,4,3,\frac{10}{3},\ \frac{8}{10/3},\ 4,\ \frac{16}{3},\frac{16}{16/5},\frac{16}{10/3},\ \frac{16}{3},4,8,8,16)$
8	16	$(1,2,2,4,3,\frac{10}{3},\ \frac{16}{5},4,4,4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
9	16	$(1,2,2,4,3,\frac{10}{3},\frac{8}{10/3},4,\frac{16}{3},4,4,8,4,8,8,16)$
10	16	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},4,4,\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
11	16	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},4,4,4,4,8,4,8,8,16)$
12	16	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\frac{16}{16/5},\frac{16}{5},4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
13	16	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\frac{16}{16/5},\frac{16}{5},\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
14	16	$(1,2,2,4,3,\frac{10}{3},\ \frac{16}{5},\frac{16}{16/5},\frac{16}{5},\ 4,\ \ 4,\ \ 8,4,8,8,16)$
15	16	$(1,2,2,4,3,\frac{10}{3},\frac{8}{10/3},4,\frac{16}{3},4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$

Table 3

	growth	Class III- Pivot Pattern
1	16	$(1,2,2,4,2,4,\ 4,\ 4,\frac{9}{2},\frac{16}{18/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
2	16	$(1,2,2,4,2,4,\ 4,\ \frac{9}{2},4,\frac{16}{18/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
3	16	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},4,4,\frac{16}{18/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
4	16	$(1,2,2,4,3,\frac{8}{3},4,4,4,\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
5	16	$(1,2,2,4,3,\frac{8}{3},4,4,4,4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
6	16	$(1,2,2,4,3,\frac{8}{3},4,4,4,4,4,8,4,8,8,16)$
7	16	$(1,2,2,4,3,\frac{8}{3},4,\frac{16}{16/5},\frac{16}{5},4,\frac{16}{8/3},\frac{16}{3},4,8,8,16)$
8	16	$(1,2,2,4,3,\frac{8}{3},4,\frac{16}{16/5},\frac{16}{5},\frac{16}{16/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$
9	16	$(1,2,2,4,3,\frac{8}{3},4,\frac{16}{16/5},\frac{16}{5},4,4,8,4,8,8,16)$
10	16	$(1,2,2,4,3,\frac{8}{3},4,4,\frac{9}{2},\frac{16}{18/5},\frac{16}{10/3},\frac{16}{3},4,8,8,16)$

Table 4

Summarising the above Tables we have the following Table for the values appearing in each class.

Pivot	1st Class	2nd Class	3rd Class	4th Class
1	1	1	1	1
2	2	2	2	2
3	2	2	2	2
4	4	4	4	4
5	2,3	2,3	2,3	2,3
6	$4,\frac{8}{3}$	$4,\frac{10}{3}$	$4,\frac{8}{3},\frac{10}{3}$	$4,\frac{10}{3}$
7	2,4	$4,\frac{8}{10/3},\frac{16}{5}$	$4,\frac{18}{5}$	$4,\frac{18}{5}$
8	4,6,8	4,5,6,8	$4,\frac{9}{2}$,5,6,8	4,5,6,8
9	$2,4,\ \frac{8}{3}$	$2,4,\frac{8}{3},\frac{16}{3},\frac{16}{5}$	$2,4,\frac{9}{2},\frac{8}{3}\ \frac{16}{5}$	$2,4,\frac{9}{2},\frac{8}{3}\ \frac{16}{5}$
10	4,8	4,5	$4,5,\frac{16}{18/5}$	$4,5,\frac{16}{18/5}$
11	4,6,8	$4,6,\frac{16}{10/3}$	$4,6,\frac{16}{10/3}$	$4,6,\frac{16}{10/3}$
12	$8,\frac{16}{3}$	$8,\frac{16}{3}$	$8,\frac{16}{3}$	$8,\frac{16}{3}$
13	4,8	4	4	4
14	8	8	8	8
15	8	8	8	8
16	16	16	16	16

Table 5

4. PIVOT STRUCTURE OF THE 20 X 20 HADAMARD MATRICES

Hall (1965) and Seberry Wallis (1972) found three equivalence classes of Hadamard matrices of order 20. After testing 3000000 equivalent matrices we found totally 1015 different pivot patterns. The following Table presents the 12 different pivot patterns attained concerning the first seven and and the last seven pivots. The intermediate pivots are varying and according to their possible values the 1015 pivot patterns were computed. The frequency of each pivot pattern concerns all the

possible values of the intermediate pivots that are appearing i.e. for the first pivot pattern of the next Table were found 125 different pivot patterns starting with the specified seven first pivots and ending with the specified seven last pivots.

growth	Pivot Pattern	Frequency
20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{20}{18/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	125
20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{20}{4},\frac{20}{8/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	195
20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{20}{16/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	123
20	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{20}{4},\frac{20}{4},\frac{20}{2},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	214
20	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{20}{4},\frac{20}{4},\frac{20}{2},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	75
20	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{20}{16/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	41
20	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{20}{18/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	17
20	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{20}{4},\frac{20}{8/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	64
20	$(1,2,2,4,2,4,4,\ldots,\frac{20}{16/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	31
20	$(1,2,2,4,2,4,4,\ldots,\frac{20}{4},\frac{20}{8/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	55
20	$(1,2,2,4,2,4,4,\ldots,\frac{20}{4},\frac{20}{4},\frac{20}{2},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	56
20	$(1,2,2,4,2,4,4,\ldots,\frac{20}{18/5},\frac{20}{10/3},\frac{20}{3},\frac{20}{4},\frac{20}{2},\frac{20}{2},20)$	19

Table 6

In none of the cases we examined was the fourth last pivot $\frac{20}{2}$. The following Table summarises the first seven and the last seven values that appear as pivots.

Pivot	1	2	3	4	5	6	7
Values	1	2	2	4	2,3	$4,\frac{10}{3}$	$4,\frac{18}{5},\frac{16}{5}$

Table 7

Pivot	14	15	16	17	18	19	20
Values	$\frac{20}{4},\frac{20}{18/5},\frac{20}{16/5}$	$\frac{20}{4},\frac{20}{10/3},\frac{20}{8/3}$	$\frac{20}{3},\frac{20}{2}$	$\frac{20}{4}$	$\frac{20}{2}$	$\frac{20}{2}$	20

Table 8

5. PIVOT STRUCTURE OF THE 32 X 32 HADAMARD MATRICES

We tested 1700000 Hadamard matrices of order 32. We found totally 414380 different pivot patterns. The following Table presents the 22 different pivot patterns attained concerning the seven first and the seven last pivots. The intermediate pivots are varying and according to their possible values the 414380 pivot patterns were computed. The frequency of each pivot pattern concerns all the possible values of the intermediate pivots that are appearing i.e. for the first pivot pattern of the next Table were found 171244 different pivot patterns starting and ending with the specified seven first and seven last pivots. We did find one case, which was equivalent to the Sylvester Hadamard matrix, which had fourth last pivot $\frac{32}{2}$.

growth	Pivot Pattern	Frequency
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{4},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	171244
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{18/5},\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	50646
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{4},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	5612
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{4},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{4},\frac{32}{2},32)$	73245
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{4},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	45681
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,10,\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	33798
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{16/5},\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	960
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{4},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	2042
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{18/5},\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	13247
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{4},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	12641
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{16/5},\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	4407
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{2},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32),$	102
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{18/5},\frac{32}{10/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	888
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{4},\frac{32}{2},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	21
32	$(1,2,2,4,2,4,4,\ldots,\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	139
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	34
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{2},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	32
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	37
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{2},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	37
32	$(1,2,2,4,3,\frac{10}{3},\frac{18}{5},\ldots,\frac{32}{4},\frac{32}{2},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	1
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{4},\frac{32}{2},\frac{32}{2},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	13
32	$(1,2,2,4,3,\frac{10}{3},\frac{16}{5},\ldots,\frac{32}{3},\frac{32}{8/3},\frac{32}{3},\frac{32}{4},\frac{32}{2},\frac{32}{2},32)$	3

Table 9

The following Table summarises the first seven and the last seven values which appear as the pivots.

Pivot	1	2	3	4	5	6	7
Values	1	2	2	4	2,3	$4,\frac{10}{3}$	$4,\frac{18}{5},\frac{16}{5}$

Table 10

Pivot	26	27	28	29	30	31	32
Values	$\frac{32}{2},\frac{32}{3},\frac{32}{4},\frac{32}{18/5},\frac{32}{16/5}$	$\frac{32}{4},\frac{32}{10/3},\frac{32}{8/3}$	$\frac{32}{3},\frac{32}{2}$	$\frac{32}{4}$	$\frac{32}{2}$	$\frac{32}{2}$	32

Table 11

6. CONCLUSIONS

The above results show that the magnitudes of a few of the first and last few pivot elements are determined. However, the sizes of the intermediate pivots vary and we believe only the last half of the pivots are directly dependent on n.

We conjecture that the fifth from last pivot can only have magnitude $\frac{n}{2}$ or $\frac{n}{3}$ and that the sixth last pivot can take the values $\frac{n}{4}$, $\frac{n}{10/3}$, or $\frac{n}{8/3}$. We also believe that the sixth pivot can have magnitude $\frac{10}{3}$ or $\frac{8}{3}$ or 4.

7. REFERENCES

CRYER, C.W. (1968): Pivot size in Gaussian elimination. Numer. Math. 12:335-345.

DAY, J. and PETERSON, B. (1988): Growth in Gaussian elimination.Amer. Math. Monthly 95:489-513.

EDELMAN, A. and FRIEDMAN, D. (1998): A counterexample to a Hadamard matrix pivot conjecture. Linear and Multilinear Algebra 44:53-56.

HALL, M. Jr. (1961): Hadamard matrices of order 16. Jet Propulsion Lab., Res. Summ., 36-10 Vol. 1, 21-26, Pasadena, CA.

HALL, M. Jr. (1965): Hadamard matrices of order 20. Jet Propulsion Lab., Technical Report, No. 32-761.

GOULD, N. (1991): On growth in Gaussian elimination with pivoting. SIAM J. Matrix Anal. Appl. 12:354-361.

KOUKOUVINOS, C., MITROULI, M. and SEBERRY, J. (2000): Growth in Gaussian elimination for weighing matrices *W(n, n-1)*, Linear Algebra and its Appl. 306:189-202.

KOUKOUVINOS, C., MITROULI, M. and SEBERRY, J. (2001): An algorithm to find formulae and values of minors of Hadamard matrices, Linear Algebra and its Appl. 330:129-147.

KOUKOUVINOS, C., MITROULI, M. and SEBERRY, J. (to appear): An infinite family of Hadamard matrices with fourth last pivot $\frac{n}{2}$, Linear and Multilinear Algebra.

MITROULI, M. and KOUKOUVINOS, C. (1996): On the growth problem for D-optimal designs, Proceedings of the First Workshop on Numerical Analysis and Applications, Lecture Notes in Computer Science, Vol. 1196, Springer Verlag, Heidelberg: 341-348.

SEBERRY WALLIS, J. (1972): Hadamard matrices, Part IV, Combinatorics: Room Squares, sum free sets and Hadamard Matrices, Lecture Notes in Mathematics, Vol 292. WALLIS, W.D., PENFOLD STREET, A. and SEBERRY WALLIS, J. (eds): Springer-Verlag, Berlin-Heidelberg-New York.

SHARPE, F. R. (1907): The maximum value of a determinant, Bull. Amer. Math. Soc. 14: 121-123.

WILKINSON, J. H. (1963): Rounding Errors in Algebraic Processes, Her Majesty's Stationery Office, London.

WILKINSON, J. H. (1988): The Algebraic Eigenvalue Problem, Oxford University Press, London.

BIOGRAPHICAL NOTES

Christos Koukouvinos received the B.S. degree in mathematics and the PhD degree in statistics, both from the University of Thessaloniki, Thessaloniki, Greece, in 1983 and 1988, respectively. In 1996, he received the Hall medal (a research award) from the ICA. He is a Fellow of the ICA and since 2001 an elected member of the Council of the ICA. He is in the Editorial Board for the Australasian Journal of Combinatorics. He is currently an Associate Professor at the National Technical University of Athens. His research interests include statistics, combinatorics, matrix theory, coding theory and computational mathematics.

Eleftherios Lappas received the B.S. degree in mathematics from the University of Athens, Athens, Greece, in 2001. He is currently a PhD student at the National Technical University of Athens in the area of computational mathematics.

Marilena Mitrouli received the B.S. degree in mathematics from the University of Athens, Greece, in 1983 and the PhD degree in computational mathematics from the City University, London, U.K., in 1991. She is currently an Assistant Professor at the University of Athens. Her research interests include numerical linear algebra, error analysis, matrix theory, and computational mathematics.

Jennifer Seberry received a BSc from the University of NSW in 1966 and a PhD in computational mathematics from La Trobe University in 1971. She is Professor and Former Head of the Department of Computer Science and Director, Centre for Computer Security Research at the University of Wollongong. In 1990 she founded the AUSCRYPT/ASIACRYPT series of International Cryptologic Conferences in the Asia/Oceania area. Her research interests include cryptoalgorithms, computer security, Hadamard matrices and computational mathematics.

Waltzing on a KDF9

Don Herbison-Evans

Faculty of Information Technology
University of Technology, Sydney
PO Box 123 Broadway, NSW 2041, Australia
don@it.uts.edu.au

This paper describes steps to use modern technology to describe the centuries old Labanotation for illustrating dance steps. These are animated by figures constructed from ellipsoids using NUDES and graphical editors. The resulting challenges are far from conquered.

Keywords: Labanotation, ellipsoid solids, general quartics, animation, graphic editors

CR Categories: D.3.2 – Specialised application languages, F.2.1 – Computation on polynomials, G.1.2 – Approximation of surfaces and contours, I.3 – Computer Graphics

1. INTRODUCTION

Early in 1973, a Sydney Choreographer, Phillipa Cullen, came to the Basser Department of Computer Science and talking to Professor John Bennett, she asked a question about a dance notation called 'Labanotation' (Hutchinson, 1954). She had a book in which were described many interesting dances, but they were listed in Labanotation (Arbeau, 1589) which she did not know. She asked the simple question: *Could a computer program be written to take the notation, and produce animated images of a synthetic figure doing the dance, for then she could copy the movements of the figure and so learn the dance?* She was brought to see me, as I was at least trying to draw human figures by computer.

Books on Labanotation and other dance notations were hard to comprehend so I started ballet lessons and later took sabbatical leave to study Labanotation with Rhonda Ryman, Drid Williams, and Ronne Arnold.

2. NUDES

The human body being curved, it is natural to try to draw it using curved primitives. The simplest of these are the quadrics. Thus the *Numerical Utility Displaying Ellipsoid Solids* (NUDES for short) was born (Herbison-Evans, 1979). As John Bennett pointed out, ellipsoids have the great advantage of only requiring the solution of a quartic equation to find where two ellipsoids intersect and where hidden lines start and end. Quartic equations can in principle be solved by fast analytic algorithms, rather than the slow iterative algorithms required for solving general polynomials.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

There are three basic algorithms published for this analytic solution to a quartic equation, each requiring the solution of a different subsidiary cubic equation. For a general quartic equation of the form:

$$x^4 + ax^3 + bx^2 + cx + d = 0$$

the subsidiary cubic can be one of the forms:

- Ferrari-Lagrange (Turnbull, 1947)

$$y^3 + by^2 + (ac - 4d)y + (a^2d + c^2 - 4bd) = 0$$

- Descartes-Euler-Cardano (Strong, 1859)

$$y^3 + (2b - 3a^2/4)y^2 + (3a^4/16 - a^2b + ac + b^2 - 4d)y + (abc - a^6/64 + a^4b/8 - a^3c/4 - a^2b^2/4 - c^2) = 0$$

- Neumark (Neumark, 1965)

$$y^3 - 2by^2 + (ac + b^2 - 4d)y + (a^2d - abc + c^2) = 0$$

It was soon found that a substantial effort was necessary to choose which algorithm to use under what circumstances (Herbison-Evans, 1995). None are completely satisfactory. It also appears to be an open problem whether there any more possible subsidiary cubics. If there are only three: why are there only three?

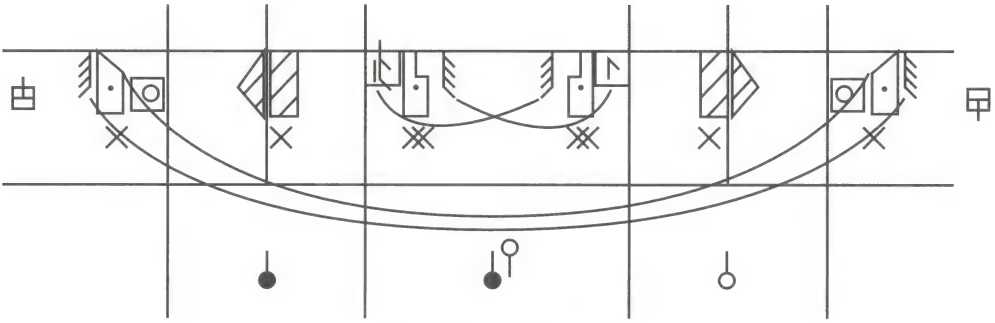
3. GRAPHICS

At that time, computing was done in Algol on the departmental English Electric KDF9. Graphic output was done at best by printing characters of various complexity on a lineprinter page. A Calcomp line plotter was acquired, which improved the resolution considerably. Animation was a rather slow process however, requiring the photographing of a series of hard copies of the individual frames with a cine camera.

The department then acquired a DEC 338 vector display. John Bennett and Doug Richardson put together a solenoid-operated 16 mm movie camera and a set of solenoid-operated colour filters, driven from the display. This made the study of movement possible for the first time. With this came the realisation that not only was the computer graphics full of computational problems, but also getting anything like a lifelike movement was going to take a great deal of study. The 338 was a difficult machine for students to program, and it was later replaced it with an Imlac PDS4. This led to a number of projects involving the economic communication of graphic data between machines (Herbison-Evans, *et al*, 1982). It also made possible "Digital Duet", the first pas-de-deux between a live dancer and a computed figure, performed at the closing ceremony of the 10th Australian Computer Conference (Herbison-Evans, *et al*, 1989).

4. DANCE NOTATION

With an interactive graphic display, it became possible to put together graphical editors for dance notation. At that time, the easiest methods for doing this were unclear, and students experimented with various menu options and other interactive techniques. A dance notation editor was a prerequisite for starting on the original *simple* problem. Several generations of students investigated the writing of a program like this, culminating in LED (Hunt, *et al*, 1989).



taking step 2 of a waltz
approximated in Labanotation

5. COLOUR

An AED512 raster colour display was then acquired for the department. Again this added a new dimension to the computing problems that needed to be solved to draw human figures dancing. Networking this to the Imlac and other departmental computers, and working out fast algorithms for doing lifelike shading to curved surfaces proved to be great student projects. (Herbison-Evans, 1980).



taking step 2
of a waltz
Fred Fortran
and Ginger
Gigabyte

6. THE WALTZ

Specifying and displaying the motion of a single dancing figure has many computational problems, but these are compounded again in any attempt to display Ballroom Dancing. The closed ballroom hold ideally requires the maintenance of five points of contact between the partners while they are dancing. For the Modern Waltz, these consist of three hand contacts:

- the man's left hand holding the lady's right hand,
- the lady's left hand resting on the top of the man's right upper arm,
- the man's right hand placed on the left shoulder blade on the back of the lady.

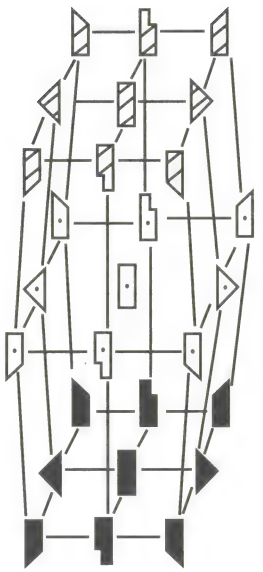
In addition to these three hand contacts, there are two more areas of contact:

- the lady's left elbow rests on the man's right elbow,
- the right area of the chest of each partner touches that of the other.

The NUDES animation system has specific commands which make this possible to arrange easily. Again the mathematical properties of quadrics are used for this. Two ellipsoids are specified as being required to be in contact using a translation or rotation of some part of a figure. One ellipsoid is transformed into a sphere, and the other (in transformed space) is moved to have its closest point to the sphere centre be at a distance equal to the radius of the sphere (Buckdale, 1983). This requires finding the smallest root to a sextic equation, the coefficients of which are derived from the parameters of the two ellipsoids.

7. LINTER

Sadly in 1975 Phillipa Cullen died, but the simple problem she raised was still an inspiration for many projects. These are now at the point of answering her simple request, with the production of LINTER: a Labanotation Interpreter This interpreter takes a file produced by LED, the Labanotation editor, and produces from it a corresponding NUDES animation script. For gesturing movements, it works by taking each Laban symbol in turn, and setting a new direction in space for the



The Labanotation symbols for the 27 major directions of space.

corresponding body part, according to which of the 27 directions in space the Laban symbol indicates.

The body part is rotated from its current direction in space to its new direction using quaternions through the period of time specified by the length of the Laban symbol. However, for locomotion, prespecified NUDES subroutines are used. This is necessary because of a difference between the specifications of a single step in normal dance movement, and a single step in Ballroom Dancing. In normal movement, the start and end of a single step is with the feet closed, but in Ballroom Dancing, a single step is most conveniently specified from open position of the feet to the next open position. The interpreter has to look ahead for "hold" signs in a support column, to ensure that the feet are not closed prematurely after a step is taken. A locomotion routine is used for a movement in each of the eight major locomotion directions of Labanotation (forward, back, left, right, and the four diagonal directions). Other NUDES routines are used to specify whether the movement is done with the left or the right leg according to the position of the Laban symbol on the notation staff.

The interpreter so far accommodates only gestural movements of the arms and legs of a single figure, and only ground support with the feet. It has serious problems with changing between heel and toe leads of the feet in locomotion, and also with the "lock" step where one foot is crossed behind or in front of the other. Nevertheless, the interpreter has already been put to use on the world-wide-web as a way for dance students to visualise notated dances. The first application involves the fifteen Championship New Vogue dances that are so popular in DanceSport competitions in Australia (Herbison-Evans, 1997).

8. CONCLUSION

The main value so far of the *Waltzing on a KDF9* project has been the number and variety of student research projects it has spawned, ranging from investigating the solution of quartic equations (Prineas, 1981) to generating animation of random Hapkido contests (Wilcockson, 1994).

Throughout their school and undergraduate careers, students are given problems to solve, the answers to which are known. Only when they study for a postgraduate degree are they offered problems for which the answers are not known. One of the main tasks of university academics is to guide their postgraduate students into this unknown, giving them the tools needed for coping with the greatest unknown of all: the future.

A more colourful version of this paper may be found at http://linus.it.uts.edu.au/~don/pubs/jmb.html

REFERENCES

ARBEAU, T. (1589): Orchesography, translated Evans, M.S,, Dover Publications 1967, New York.

BUCKDALE, R.S. (1983): Normal feet for dancing ellipsoids. *Proceedings Graphics83*, HVISTENDAHL, H. (ed). Sydney: 71-76.

HERBISON-EVANS, D. (1979): A human movement language for computer animation. *In Language Design and Programming Methodology*. TOBIAS, J.L.(ed). Springer-Verlag: 117-128.

HERBISON-EVANS, D. (1980): Rapid raster ellipsoid shading. *Computer Graphics* 13(4): 355-361.

HERBISON-EVANS, D., NICHOLSON, T. and BERRY, C.B. (1982): The transmission and drawing of elliptical arcs.

Proceedings 9th Australian Computer Conference: 497-502.
HERBISON-EVANS, D., ADAMS, T. and KEIGHERY, K. (1989): Digital duet. Basser Department of Computer Science, Technical Report 338. University of Sydney.
HERBISON-EVANS, D. (1995): Solving quartics and cubics for graphics. In *Graphics Gems V*. PAETH, A.(ed). Academic Press, Chestnut Hill:3-15.
HERBISON-EVANS, D. (1997): Australian new vogue dancing on the web. O'BRIEN, W., GRASBY, D., BRIGG, M. and HUNGERFORD. L.(ed). Central Queensland University, Rockhampton: 189-192.
HUNT, F.E.S., POLITIS, G. and HERBISON-EVANS, D. (1989): An interactive graphical editor for labanotation., Basser Department of Computer Science, Technical Report 343. University of Sydney.
HUTCHINSON, A. (1954): Labanotation: The system for recording movement. Theatre Arts Books, New York.
NEUMARK, S. (1965): Solution of cubic and quartic equation., Pergamon Press, Oxford.
PRINEAS, C. (1981): Algorithms for solving quartic equations. Honours attachment report. Basser Department of Computer Science, University of Sydney.
STRONG, T. (1859): Elementary and higher algebra. Pratt, Oakley & Company, New York. p. 469.
TURNBULL, H.W. (1947): Theory of equations. Oliver and Boyd, London, Fourth Edition. p. 130.
WILCOCKSON, A.J. (1994): MATS – Martial arts teaching system. Honours Project Report. Basser Department of Computer Science, University of Sydney.

BIOGRAPHICAL NOTES

Don Herbison-Evans has a BA, MA and DPhil from Oxford and a Diploma in Dance Education. After a variety of careers he became an academic in the Basser Department of Computer Science at the University of Sydney. He is now a part time academic and Honorary Associate in the Faculty of Information Technology at the University of Technology, Sydney. He has a wide range of interests but his abiding academic obsession has been the design and study of languages for the description and prescription of human movement. Deciding that dancers have been designing sytems for communicating movement for centuries, he has been learning and studying many forms of dance.

Screw-Based Computer Graphics

I Parkin

School of Information Technologies
The University of Sydney, NSW 2006, Australia
ian@cs.usyd.edu.au

This paper provides expressions for the line and certain key points of the image of a kinematic screw as it is seen drawn in perspective on a computer display screen. Rather than apply computer-graphic projections to arbitrarily chosen points of the screw, the expressions employ screw-theoretic measures which are obtained directly from the screw. Computational procedures for evaluating these measures are likely to exist amongst software tools used for other kinematic analysis.

Keywords: screw-theoretic, computational kinematics, computer-aided geometrical design, Plucker and Study coordinates, eye-frames and view planes.

CR Categories: I.3.5 – Computational Geography and Object Modelling.

1. INTRODUCTION

Situations often arise within *computational kinematics* [Parkin *et.al.*, 2000] and *computer-aided geometrical design* [Sprott *et.al.*, 1997] which call for computer-graphical display of kinematic artefacts such as the lines of action of *wrenches* or *screw velocities*, or the *screw axes* of displacements, finite or infinitesimal. In each of these, the need arises to convert from measures like the *Plucker* or *Study* coordinates used in specifying *screws* and *biquaternions* to the real two- and three-vector representations of *points* which are the usual currency of computer graphics.

This paper applies screw-theoretic methods to a screw to derive direct expressions for on-screen points and lines of its image, seen in perspective. These expressions employ screw-theoretic measures and can be readily implemented by use of computational procedures of a kind which are likely to exist amongst the software tools used for other kinematic analysis.

2. NOTATION AND BASIC GEOMETRY

Definition of a Screw

A *screw* will be written as a 3-vector of dual numbers

$$\left.\begin{array}{l} \hat{\mathbf{S}} = \mathbf{S} + \varepsilon\,\mathbf{S_p} = |\hat{\mathbf{S}}|\,(1+\varepsilon\,p)\,\hat{\mathbf{s}}, \quad \hat{\mathbf{s}} = \mathbf{l} + \varepsilon\,\mathbf{M}, \\ \hat{\mathbf{s}}^2 = \mathbf{l}^2 + \varepsilon\,2\,\mathbf{l}\cdot\mathbf{M} = 1 + \varepsilon\,0, \quad \mathbf{l}\times\mathbf{M} = \mathbf{R}, \end{array}\right\} \tag{1}$$

in which ε is a quasi-scalar such that $(a+\varepsilon\,b = c+\varepsilon\,d) \Leftrightarrow (a = c) \wedge (b = d)$ for all real a, b, c, d, and satisfying $\varepsilon^2 = 0$. We use emboldened letters to represent 3-vectors, lower case

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

letters indicating *unit* vectors and the overwritten 'hat' symbol indicating dual quantities. $|\hat{\mathbf{S}}|$ is the real *magnitude* and p is the real *pitch* of the screw $\hat{\mathbf{S}}$, and $\hat{\mathbf{s}}$ is its *normalised line* which, regarded as a screw in its own right, has unit magnitude and zero pitch. The screw $\hat{\mathbf{S}}$ is spatially located by its *direction vector* $\mathbf{S}$ and *moment vector* $\mathbf{S_p}$; more specifically, the line $\hat{\mathbf{s}}$ of the screw has the direction of the vector of direction cosines $\mathbf{l}= (l,\ m,\ n)$ and its *origin-radius* vector $\mathbf{R}$ is determined by the moment vector $\mathbf{M}= (P,\ Q,\ R)$ which is orthogonal to $\mathbf{l}$. The real values $l,\ m,\ n,\ P,\ Q,\ R$ are the *Plucker coordinates* of that line. For any two screws

$$\hat{\mathbf{A}}= \mathbf{A}+\varepsilon\,\mathbf{A_p}= \mathbf{A}+\varepsilon\,(p_A\,\mathbf{A}+\mathbf{A_0}) \quad \textit{and} \quad \hat{\mathbf{B}}= \mathbf{B}+\varepsilon\,\mathbf{B_p}= \mathbf{B}+\varepsilon\,(p_B\,\mathbf{B}+\mathbf{B_0})$$

of pitches p_A, p_B, their *scalar product* is the dual number given by

$$\hat{\mathbf{A}}\cdot\hat{\mathbf{B}}= \mathbf{A}\cdot\mathbf{B}+\varepsilon\,\hat{\mathbf{A}}\otimes\hat{\mathbf{B}} \quad \textit{where} \quad \hat{\mathbf{A}}\otimes\hat{\mathbf{B}}= \mathbf{A}\cdot\mathbf{B_p}+\mathbf{B}\cdot\mathbf{A_p}$$

in which $\hat{\mathbf{A}}\otimes\hat{\mathbf{B}}$ is the *mutual moment* of the screws $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$, also expressed by

$$\hat{\mathbf{A}}\otimes\hat{\mathbf{B}}= \hat{\mathbf{A}}\otimes_0\hat{\mathbf{B}}+(p_A+p_B)\,\mathbf{A}\cdot\mathbf{B} \quad \textit{where} \quad \hat{\mathbf{A}}\otimes_0\hat{\mathbf{B}}= \mathbf{A}\cdot\mathbf{B_0}+\mathbf{B}\cdot\mathbf{A_0} \tag{2}$$

in which $\hat{\mathbf{A}}\otimes_0\hat{\mathbf{B}}$ is the *mutual moment* of the *lines of the screws* $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$. Two screws $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ *intersect* if $\hat{\mathbf{A}}\otimes_0\hat{\mathbf{B}}= 0$. They are said to be *perpendicular* if $\mathbf{A}\cdot\mathbf{B}= 0$, to be *reciprocal* if $\hat{\mathbf{A}}\otimes\hat{\mathbf{B}}= 0$, and to be *orthogonal* if both of these are true, *i.e.* if $\hat{\mathbf{A}}\cdot\hat{\mathbf{B}}= 0$, which implies that each intersects the other at right angles. Their *cross product* screw $\hat{\mathbf{A}}\times\hat{\mathbf{B}}$ is sited in the *common perpendicular line* of $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$.

Axial Frames

We represent the typical right-handed reference frame by orthogonal normalised axial lines $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, $\hat{\mathbf{z}}_i$ for which

$$\left.\begin{array}{l} \hat{\mathbf{x}}_i\cdot\hat{\mathbf{y}}_i= \hat{\mathbf{y}}_i\cdot\hat{\mathbf{z}}_i= \hat{\mathbf{z}}_i\cdot\hat{\mathbf{x}}_i= 0,\ \ \hat{\mathbf{x}}_i^2= \hat{\mathbf{y}}_i^2= \hat{\mathbf{z}}_i^2= 1, \\ \hat{\mathbf{x}}_i\times\hat{\mathbf{y}}_i= \hat{\mathbf{z}}_i,\ \ \hat{\mathbf{y}}_i\times\hat{\mathbf{z}}_i= \hat{\mathbf{x}}_i,\ \ \hat{\mathbf{z}}_i\times\hat{\mathbf{x}}_i= \hat{\mathbf{y}}_i. \end{array}\right\} \tag{3}$$

The point $\mathbf{V}_i$ in which such mutually orthogonal lines meet is, of course, the origin of this i-frame. Each axis, since it contains the point $\mathbf{V}_i$, may be written

$$\hat{\mathbf{x}}_i\equiv \mathbf{x}_i+\varepsilon\,\mathbf{x}_{0i}= \mathbf{x}_i+\varepsilon\,\mathbf{V}_i\times\mathbf{x}_i, \quad \textit{etc.} \tag{4}$$

so the origin of such an orthonormal triad can always be identified in the form

$$\mathbf{V}_i= \mathbf{y}_{0i}\cdot\mathbf{z}_i\,\mathbf{x}_i\ +\ \mathbf{z}_{0i}\cdot\mathbf{x}_i\,\mathbf{y}_i\ +\ \mathbf{x}_{0i}\cdot\mathbf{y}_i\,\mathbf{z}_i.$$

Frame Transformation

If – in some common reference system – we know the coordinates $\hat{\mathbf{S}}= \mathbf{S}+\varepsilon\,\mathbf{S_p}$ of a general screw and of the i-frame axes $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, and $\hat{\mathbf{z}}_i$, we may always *transform* that screw into i-frame coordinates by writing

$$\hat{\mathbf{S}}_i\equiv \mathbf{S}_i+\varepsilon\,\mathbf{S}_{\mathbf{p}_i}= \begin{bmatrix} \hat{\mathbf{x}}_i^T \\ \hat{\mathbf{y}}_i^T \\ \hat{\mathbf{z}}_i^T \end{bmatrix}\hat{\mathbf{S}}= \begin{bmatrix} \mathbf{x}_i\cdot\mathbf{S} \\ \mathbf{y}_i\cdot\mathbf{S} \\ \mathbf{z}_i\cdot\mathbf{S} \end{bmatrix}+\varepsilon\begin{bmatrix} \hat{\mathbf{x}}_i\otimes\hat{\mathbf{S}} \\ \hat{\mathbf{y}}_i\otimes\hat{\mathbf{S}} \\ \hat{\mathbf{z}}_i\otimes\hat{\mathbf{S}} \end{bmatrix} \tag{5}$$

in which the first-written matrix (on the left) is shown by equations (3) to be a 3×3 dual orthogonal matrix. The inverse transformation is achieved by the transpose of that matrix, whose columns are the dual 3-vectors $\hat{\mathbf{x}}_i$, $\hat{\mathbf{y}}_i$, $\hat{\mathbf{z}}_i$.

3. POINTS, PLANES AND LINES

Let us denote the general *plane* by the *normal-point* bivector $[\mathbf{N},\ \mathbf{P}]$ in which the 3-vector $\mathbf{N}$, of any non-zero magnitude, is the normal direction and the 3-vector $\mathbf{P}$ is some particular point of the plane. Then the general point $\mathbf{Q}$ of the plane $[\mathbf{N},\ \mathbf{P}]$ is characterised by the fact that its vector difference $\mathbf{P}-\mathbf{Q}$ from point $\mathbf{P}$ is orthogonal to the normal $\mathbf{N}$, so we have

$$(\mathbf{P}-\mathbf{Q}) \cdot \mathbf{N} = 0 \qquad i.e. \qquad \mathbf{Q} \cdot \mathbf{N} = \mathbf{P} \cdot \mathbf{N}, \tag{6}$$

which is the *equation of the plane*. The origin distance of the plane is $\mathbf{P} \cdot \mathbf{N} / |\mathbf{N}|$.

Line of Intersection of Two Planes

For two generally disposed planes $[\mathbf{N}_A,\ \mathbf{P}_A]$ and $[\mathbf{N}_B,\ \mathbf{P}_B]$, we now determine their line of intersection – which is a line at infinity if the planes are parallel – as a zero-pitched screw $\hat{\mathbf{L}}_{AB} = \mathbf{L}_{AB} + \varepsilon \mathbf{L}_{\mathbf{P}AB}$, with $\mathbf{L}_{AB} \cdot \mathbf{L}_{\mathbf{P}AB} = 0$, and such that $\hat{\mathbf{L}}_{BA} = -\hat{\mathbf{L}}_{AB}$. The direction component of this line, being perpendicular to both normals $\mathbf{N}_A$ and $\mathbf{N}_B$, is conveniently expressed as

$$\mathbf{L}_{AB} = -\mathbf{L}_{BA} = \mathbf{N}_A \times \mathbf{N}_B. \tag{7}$$

For any point $\mathbf{Q}$ of the line of intersection, in both planes, equation (6) yields

$$\mathbf{Q} \cdot \mathbf{N}_A = \mathbf{P}_A \cdot \mathbf{N}_A \quad and \quad \mathbf{Q} \cdot \mathbf{N}_B = \mathbf{P}_B \cdot \mathbf{N}_B,$$

so equation (1) shows the moment component of the line $\hat{\mathbf{L}}_{AB}$ to be

$$\mathbf{L}_{\mathbf{P}AB} = \mathbf{Q} \times \mathbf{L}_{AB} = \mathbf{Q} \cdot \mathbf{N}_B \mathbf{N}_A - \mathbf{Q} \cdot \mathbf{N}_A \mathbf{N}_B = \mathbf{P}_B \cdot \mathbf{N}_B \mathbf{N}_A - \mathbf{P}_A \cdot \mathbf{N}_A \mathbf{N}_B,$$

which is a linear combination of the normals $\mathbf{N}_A$ and $\mathbf{N}_B$ in which the scalar products are coefficients. Thus, in summary, the directed line of intersection is

$$\hat{\mathbf{L}}_{AB} = -\hat{\mathbf{L}}_{BA} = \mathbf{N}_A \times \mathbf{N}_B + \varepsilon (\mathbf{P}_B \cdot \mathbf{N}_B \mathbf{N}_A - \mathbf{P}_A \cdot \mathbf{N}_A \mathbf{N}_B). \tag{8}$$

We observe that as the normal directions $\mathbf{N}_A$ and $\mathbf{N}_B$ approach parallelism,

$$\mathbf{N}_A \to |\mathbf{N}_A| \mathbf{n} \quad and \quad \mathbf{N}_B \to |\mathbf{N}_B| \mathbf{n},$$

the line of intersection remains well-behaved and approaches the line-at-infinity

$$\hat{\mathbf{L}}_{AB} = -\hat{\mathbf{L}}_{BA} = |\mathbf{N}_A| |\mathbf{N}_B| [\mathbf{0} + \varepsilon (\mathbf{P}_B - \mathbf{P}_A) \cdot \mathbf{n}\, \mathbf{n}].$$

4. REFERENCE FRAMES AND DRAWING WINDOWS

We postulate a World whose reference axes are the self-defining lines

$$\hat{\mathbf{x}} = (1,\ 0,\ 0) + \varepsilon \mathbf{0}, \quad \hat{\mathbf{y}} = (0,\ 1,\ 0) + \varepsilon \mathbf{0}, \quad \hat{\mathbf{z}} = (0,\ 0,\ 1) + \varepsilon \mathbf{0}. \tag{9}$$

The typical screw $\hat{\mathbf{S}}$ of this World consists of a linear combination of those axes, *viz.*

$$\hat{\mathbf{S}} = \mathbf{S} + \varepsilon \mathbf{S_p} = \hat{L}\,\hat{\mathbf{x}} + \hat{M}\,\hat{\mathbf{y}} + \hat{N}\,\hat{\mathbf{z}}. \tag{10}$$

whose World coordinates $\hat{L}$, $\hat{M}$, and $\hat{N}$ are general dual numbers. Amongst those screws there are particular triads of orthonormal lines $\hat{\mathbf{x}}_A$, $\hat{\mathbf{y}}_A$, $\hat{\mathbf{z}}_A$, satisfying equations (3) which provide reference frames for particular artefacts – representing solid bodies, clipping surfaces, *etc.* – which exist within the World. We suppose these artefacts to be manipulable, in form and World location, through user interaction by means of input devices.

A special artefact is the *viewing system* (see Figure 1) which consists of:

- a right-handed triad of lines $\hat{\mathbf{u}}_V$, $\hat{\mathbf{v}}_V$, $\hat{\mathbf{n}}_V$ which constitutes the *view reference frame*, or *view-frame*,

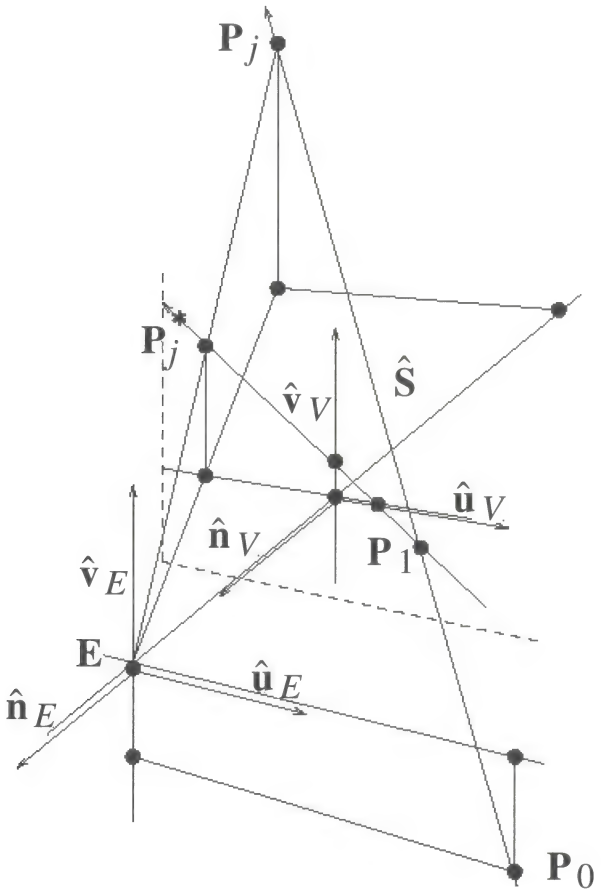


Figure 1: Showing the right-handed triad of orthonormal lines $\hat{\mathbf{u}}_V$, $\hat{\mathbf{v}}_V$, $\hat{\mathbf{n}}_V$ which constitute the *view reference frame*. Images of 3-dimensional objects are drawn in perspective, as seen by the *eye* E, on the *view-plane* which lies in the $\hat{\mathbf{u}}_V\,\hat{\mathbf{v}}_V$-plane. For example, the screw $\hat{\mathbf{S}}$ passing through points $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_j$ is depicted by the line of that plane which passes through the points $\mathbf{P}_1$, $\mathbf{P}_j^*$. The *eye-frame*, of orthonormal axes $\hat{\mathbf{u}}_E$, $\hat{\mathbf{v}}_E$, $\hat{\mathbf{n}}_E$, parallel with the former, which meet in the *eye* E, is a convenient alternative frame for analysis.

- the *eye*, which lies at point $\mathbf{E}$ on the axis $\hat{\mathbf{n}}_V$ at distance e from the view-frame origin,
- the *view-plane*, being the $\hat{\mathbf{u}}_V\hat{\mathbf{v}}_V$-plane, on which depictions of other artefacts are drawn in perspective, as seen from the eye; in this role, the view-plane represents the screen of a computer terminal.

Our *view-plane* captures the computer-graphical notion of a *window*, as a plane located within the World (without invoking the additional apparatus of its bounding rectangle). In considering drawings of artefacts to be made directly on the view-plane, in coordinates u, v measured on the respective axes $\hat{\mathbf{u}}_V$, $\hat{\mathbf{v}}_V$, we omit mention of the simple two-dimensional translation and scaling between these *window* coordinates and the *viewport* coordinates of a screen which are needed in use of an actual computer terminal. Following standard viewing conventions, we consider the coordinates u, v to be respectively measured horizontally to the right, and vertically upward, on the screen.

It is often convenient to conduct analysis in the *eye-frame* which consists of a triad of lines known in World coordinates as $\hat{\mathbf{u}}_E$, $\hat{\mathbf{v}}_E$, and $\hat{\mathbf{n}}_E$ which intersect at the *eye* $\mathbf{E}$ as origin and such that $\hat{\mathbf{u}}_E$, $\hat{\mathbf{v}}_E$ are respectively parallel to $\hat{\mathbf{u}}_V$, $\hat{\mathbf{v}}_V$ while $\hat{\mathbf{n}}_E$ is identical with $\hat{\mathbf{n}}_V$. Coordinates u and v are, of course, measured identically in both of these frames.

In either of these frames, we distinguish the general point $\mathbf{P}=\ (u,\ v,\ n)$ from its perspective image which we denote by $\mathbf{P}^*=\ (u^*,\ v^*,\ n^*)$.

5. PERSPECTIVE PROJECTION OF A LINE

The geometry of perspective dictates that any screw $\hat{\mathbf{S}}$ should be depicted by the *line of intersection* of the *view-plane* with the *plane containing the screw and the eye* (see Figure 1). In identifying this line by use of equation (8), we work in the *eye-frame* where these planes are easily identified. The view-plane, with normal (0, 0, 1) at distance $-e$ from the eye in the $\hat{\mathbf{n}}_E$-direction, can be written immediately as

$$[\mathbf{N}_A,\ \mathbf{P}_A]=\ [(0,\ 0,\ 1),\ (0,\ 0,\ -e)]. \tag{11}$$

Of the other plane we observe, since it contains the origin (at the eye), that its normal has the direction of the moment component of any line it contains, provided that such a line does not itself contain the eye. Now, by equation (5), the *line* of the screw denoted in the World by $\hat{\mathbf{S}}$ is known in the eye-frame as

$$\hat{\mathbf{L}}\equiv\ \mathbf{L}+\varepsilon\,\mathbf{M}=\ (\mathbf{S}\cdot\mathbf{u}_E,\ \mathbf{S}\cdot\mathbf{v}_E,\ \mathbf{S}\cdot\mathbf{n}_E)+\varepsilon\,(\hat{\mathbf{S}}\otimes_0\hat{\mathbf{u}}_E,\ \hat{\mathbf{S}}\otimes_0\hat{\mathbf{v}}_E,\ \hat{\mathbf{S}}\otimes_0\hat{\mathbf{n}}_E). \tag{12}$$

If this line contains the eye, the moment part $\mathbf{M}$ is null, and the screw $\hat{\mathbf{S}}$ – seen 'end-on' – is represented on the view-plane merely by the point

$$(u^*,\ v^*)=\ \frac{-e}{\mathbf{S}\cdot\mathbf{n}_E}\,(\mathbf{S}\cdot\mathbf{u}_E,\ \mathbf{S}\cdot\mathbf{v}_E), \tag{13}$$

Otherwise, the plane which is common to the line $\hat{\mathbf{L}}$ and the eye may be denoted

$$[\mathbf{N}_B,\ \mathbf{P}_B]=\ [\mathbf{M},\ (0,\ 0,\ 0)]=\ [(M_u,\ M_v,\ M_n),\ (0,\ 0,\ 0)] \tag{14}$$

where, for brevity, we have written $(M_u,\ M_v,\ M_n)\equiv\ (\hat{\mathbf{S}}\otimes_0\hat{\mathbf{u}}_E,\ \hat{\mathbf{S}}\otimes_0\hat{\mathbf{v}}_E,\ \hat{\mathbf{S}}\otimes_0\hat{\mathbf{n}}_E)$. On sub-

stituting the planes of expressions (11) and (14) into equation (8), we find that the image of $\hat{\mathbf{S}}$ – to be drawn on the screen with the same directed sense as $\hat{\mathbf{S}}$ – is the line expressed in the *eye-frame* by

$$\hat{\mathbf{L}}_{AB} = \mathbf{M} \times (0,\ 0,\ 1) - \varepsilon\, e\, \mathbf{M} = (M_v,\ -M_u,\ 0) - \varepsilon\, e\, (M_u,\ M_v,\ M_n) \tag{15}$$

and – when transformed to the parallel axes $\hat{\mathbf{u}}_V$ and $\hat{\mathbf{v}}_V$ of the *view-frame* – in the plane of the screen by

$$\hat{\mathbf{L}}'_{AB} = (M_v,\ -M_u,\ 0) - \varepsilon\, (0,\ 0,\ e\, M_n) = (\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{v}}_E,\ -\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{u}}_E,\ 0) - \varepsilon\, (0,\ 0,\ e\, \hat{\mathbf{S}} \otimes_0 \hat{\mathbf{n}}_E). \tag{16}$$

Thus it is an effect of the perspective transformation that the non-zero coordinates of the image line on the *view*-frame axes $\hat{\mathbf{u}}_V$, $\hat{\mathbf{v}}_V$, $\hat{\mathbf{n}}_V$ consist only of the mutual moments of the line of $\hat{\mathbf{S}}$ with the *eye*-frame axes $\hat{\mathbf{u}}_E$, $\hat{\mathbf{v}}_E$, $\hat{\mathbf{n}}_E$. These mutual moments may be evaluated in *any* frame. In particular, they may be evaluated – without any need for transformation – in the fundamental World frame where the axial lines $\hat{\mathbf{u}}_E$, $\hat{\mathbf{v}}_E$, $\hat{\mathbf{n}}_E$ and the typical $\hat{\mathbf{S}}$ are typically specified.

The image line makes $\hat{\mathbf{u}}_V$- and $\hat{\mathbf{v}}_V$-axis intercepts, and has slope on the screen, given respectively by

$$u^*{}_{v=0} = e\, \frac{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{n}}_E}{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{u}}_E},\quad v^*{}_{u=0} = e\, \frac{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{n}}_E}{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{v}}_E} \quad \textit{and} \quad \frac{dv^*}{du^*} = -\frac{v^*{}_{u=0}}{u^*{}_{v=0}} = -\frac{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{u}}_E}{\hat{\mathbf{S}} \otimes_0 \hat{\mathbf{v}}_E}, \tag{17}$$

of which the first two values show expected proportionality with the distance e from the screen to the eye. In seeking end-points in the screen between which to draw the line, we may solve for its intersections with any chosen *clipping* lines. *E.g.*, its intersections with an axially-oriented rectangular boundary in the view-plane are easily found by observing that the image line meets any vertical line of abscissa U and any horizontal line of ordinate V at the ordinate and abscissa values given respectively by

$$v^*{}_{u=U} = v^*{}_{u=0} + \frac{dv^*}{du^*}\, U \quad \textit{and} \quad u^*{}_{v=V} = u^*{}_{v=0} + \frac{du^*}{dv^*}\, V.$$

6. CONCLUSION

Expressions have been derived for the line and certain key points of the image of a kinematic *screw* as it is seen drawn in perspective on a computer display screen. The expressions employ screw-theoretic measures obtained directly from the screw and can be applied immediately by use of computational procedures for evaluating these measures which are likely to exist amongst software tools used for standard kinematic analysis.

7. ACKNOWLEDGEMENT

The author acknowledges the assistance of the School of Information Technologies, The University of Sydney, in preparation of this paper.

REFERENCES

SPROTT, K. and RAVANI, B. (1997): Ruled surfaces, lie groups, and mesh generation. *Proceedings of the 1997 ASME Design Engineering Technical Conference* (CD-ROM), DETC97/DAC-3996, American Society of Mechanical Engineers. 1-9.

PARKIN, I. and PRESTON, J. (2000): Analysis of the Bennett Mechanism by means of finite displacement screws. *Proceedings of a Symposium Commemorating the Legacy, Works, and Life of Sir Robert Stawell Ball*, Trinity College, University of Cambridge, July.

Strong and Weak Memory Consistencies

Chung-Kwong Yuen
School of Computing
National University of Singapore
email: yuenck@comp.nus.edu.sg

A scheme to combine the benefits of strong and weak memory consistency systems is suggested, by embedding Linda-like tuple locks in shared objects to act as control indicators.

Keywords: strong and weak memory consistency, Linda tuple locks,
CR Categories: B.3.2 – Cache memories, Shared memory, D.4.1 – Concurrency, Deadlocks.

1. INTRODUCTION

The issue of memory consistency arises whenever multiple copies of shared information are handled by separate processes. Specifically, the question amounts to: when will modifications made by one process become visible to other processes? If the answer is "immediately", then we have what is termed "sequential memory consistency". The cache coherence protocol adopted by shared memory multiprocessors is an example of such a memory consistency mechanism, with write operations in one cache causing invalidation of copies of the same memory block in other caches, hence causing future reads on such locations to obtain fresh copies that contain the latest modifications.

Sequential memory consistency is considered to be a "strong" consistency, and it is quite costly to provide, particularly on distributed systems where the runtime system has to enforce it between multiple copies held in the memories of different processors; weaker memory consistency models, which retreat from a guarantee of immediate visibility of writes on one copy to the other copy holders, aim to improve efficiency without hampering the effectiveness of parallel programming systems. By delaying the forwarding of write results to readers until certain "strategic moments", multiple modifications are buffered and consolidated, so that only the final result is forwarded once, rather than multiple modifications being forwarded individually as they occur.

For example, a process would lock a shared page, perform a number of writes on it, and unlock, without other processes becoming aware of modifications having been made on the shared information; but when another process requests a lock on the same page, it would be notified of the changes and would then acquire the information through a standard system mechanism. The cost of memory sharing may be further reduced by exploiting the virtual memory mechanism: when a lock on a group of pages is acquired, instead of forwarding new page contents, the system merely asks the process that acquires the lock to invalidate its own shared pages, so that new contents are forwarded only when a page is actually accessed, while avoiding transferring pages that are not actually used.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

Since there are different choices of the "strategic moments" and different methods for buffering and consolidating write results, varieties of weak memory consistency models are possible, and much research has gone into studies of their characteristics and performance. In all these schemes, the objective is to allow a shared item to be referred to using the same virtual address on each processor; the maintenance of various local copies consistent with each other is achieved by the runtime system transparently to the programmer. This allows a distributed system to behave as if it were a shared memory system. However, it does not necessarily mean that any shared-memory parallel program can run without modification on such a system, as the existence of multiple copies, which may differ from each other at times, requires the programmer to know the exact rules under which consistency is guaranteed; a program written on a real shared memory system, with coherent caches and single copies of shared information in memory, may not have observed the same rules.

The present paper will look at an additional issue, suggesting the desirability of adding "strength" to weak memory consistency by using "content rich" locks, more specifically, Linda tuples embedded in shared buckets or objects.

A comprehensive survey of distributed shared memory systems is given in (Keleher).

2. THE NEED FOR BETTER LOCKS

A parallel programming system may provide just a single, parameterless lock for use in all situations. With this, process X entering critical region A would exclude process Y from entering critical region B, even if A and B do not conflict. This is termed false exclusion. Similarly, if a group of objects resides within a shared memory block protected by a lock, then concurrent accesses on different objects would be mutually exclusive even though there is no actual content sharing. This is termed false sharing.

To minimise these problems, it is desirable to use locks that specify, either specifically or by implication from the execution context, what is being shared. However, while making such specifications more detailed might reduce false sharing, it would increase the cost of maintaining locks; not only do multiple locks take up more resources, each lock has to have a wait queue to store unsuccessful lock requests, which are to be reactivated when a lock is released. The two factors, system simplicity versus application program efficiency, need to be balanced.

It may also be desirable to specify "why" in addition to "what", such as having separate read and write locks, since a writer must exclude readers and other writers while it is writing, whereas multiple readers can read concurrently. Lock requests may also be conditional; for example, locking a stack object to perform a pop should only succeed if the stack is non-empty. Finally, locking could be associated with copying: if a request to lock X under condition Y is successful, then the content of X is sent to the requesting process, with the new content produced being returned to the shared information pool when unlocking occurs.

The Linda tuplespace method in effect provides a dynamic pool of individually lockable objects, with locking conditions specified by search keys. For example, a stack tuple would have an empty/notempty field in addition to the stack identifier, and a pop request would require a tuple In operation using (Stack, NotEmpty) as search key. The In succeeds if the key matches; if the stack is currently empty, its tuple does not match the In search key and the requesting process suspends, until some other process obtains the stack tuple, pushes new content onto the stack, and releases the tuple, with NotEmpty instead of Empty in the second search key field. Depending on the implementation, a successful In operation may cause the stack pointer or the stack content itself to be given to the requesting process.

In such an arrangement, an In request can be initially unsuccessful, and is buffered so that it can match a tuple created or returned to the tuplespace at a later time; in other words, because both writes and reads are buffered, Linda provides an even stronger memory consistency than sequential consistency, in which readers see the results of past writes but, unlike a Linda In, no future ones. Since a programmer can use tuple read and write operations without worrying about timing, he is relieved of some of the burden of program correctness, which is transferred to the parallel runtime system in the form of a correct tuple management sub-system. In other words, a stronger memory consistency system seeks to give the programmer better information sharing support, but at a cost.

The buffering of both unused tuples and unsuccessful requests means that a new tuple causes a search operation trying to match requests already in the pool, just as a new request causes a search attempting to match an existing tuple. In a distributed system with a central pool, there can be a large number of such searches carried out in a large pool, which would be expensive. While the need to improve efficiency by distributing the tuplespace is recognized, there is no simple, generally agreed scheme for such distributions.

A comprehensive though somewhat dated introduction to Linda is given in (linda).

3. EMBEDDED TUPLES

We now suggest the idea of tuples within shared memory blocks (which we call "buckets") or shared objects. Processes sharing a bucket may refer to tuples in it using BucketID.TupleID, and a ReadTuple request may specify a search key in addition to the ID; since the location is uniquely defined by the ID, the key requires a matching operation rather than a search of a shared pool, and a matching failure causes the request to be queued. When a tuple is modified, the queued requests need to be searched to see if there is now a match; hence, searching may still be necessary, but is now confined to a particular tuple's request queue only.

A tuple access may modify as well as retrieve its content; once this occurs, another request with the same key would no longer match and would suspend. This corresponds to locking. A second modify returning the tuple key to its previous value would result in unlocking. Combining the locking/unlocking with copying of shared memory block results in the InBucket (or InObject) and OutBucket (or OutObject) operations, which, like ReadTuple, specify a tuple ID, matching key, and possibly tuple modifications. An In causes the latest content of the shared block to be transferred to a process's local copy, possibly by just invalidating the local copy to cause subsequent transfer of actually used pages. An Out makes the modified local copy visible to other processes, replacing the previous shared copy content by overwriting, invalidation or just the sending of changes (diffs).

The last option permits a variety of multi-writer protocols. For example, if a writer executes an InBucket without modifying the tuple matching key, then other writers using the same key can concurrently execute InBucket, thus refreshing their own copies; OutBucket operations then cause each process's diffs to be consolidated into the shared copy. The tuple must contain ReaderCount and WriterCount fields, which are incremented and decremented by readers/writers as they execute In/Out, to ensure that no reader starts while writing is taking place, and no diff consolidation occurs while the old copy still has readers. Note that it is safe for writers to start while reading is taking place, since a writer has its own copy, but a writer cannot "finish" until the previously started reads have ended so that the old copy can be replaced by a new, consistent copy. In other words, the reader-writer algorithm for a distributed shared memory system is not identical to the reader-writer algorithm for real shared memory systems, where writers cannot "start" till all the current readers have finished.

The embedded tuples scheme seeks to combine the use of strong and weak memory consistencies: strong for the tuples, but weak for the shared buckets/objects. While the cost of using tuples, despite the simplification due to the fixing of tuple locations, may still be significant, it is now incurred as part of the overhead for sharing larger blocks of information and hence is less burdensome.

The scheme outlined above has been given a preliminary implementation as an MSc project (linda; Sy, 2001) using PVM as the underlying transfer mechanism. The results generally indicate that application programs using the embedded tuple scheme are somewhat less efficient than those directly coded in PVM but are simpler to write. A more comprehensive implementation will be initiated in 2002 as a PhD project. Additional features of tuple locks relating to the caching and encryption of shared data will also be considered.

REFERENCES

KELEHER: www.cs.umd.edu/~keleher/bib/dsmbiblio/dsmbiblio.html
LINDA: www.qpsf.edu.au/workshop/linda/linda.html
SY, G. (2001): MSc Thesis, School of Computing, National University of Singapore.
SY, G. (2002): A C++/Tuple-Lock Implementation for Distributed Objects. Annual Review of Scalable Computing 4:152-181.

BIOGRAPHICAL NOTES

Chung-Kwong Yuen has been with the Department of Computer Science, National University of Singapore since 1983, and was department chair from 1985 to 1993. He received his PhD from Basser Department of Computing in 1973, and worked at Australian National University, University of Tasmania and University of Hong Kong before moving to Singapore. He has also been a visiting scholar at CSIRO, University of Texas and Rice University.

A Logic MiniTrainer

Allan G Bromley

Honorary Associate
School of Information Technologies, University of Sydney
allan@it.usyd.edu.au

Students learn by doing. This paper describes the teaching of logic, simple computer architecture and its associated microcode system through a series of hardware logic trainers.

Keywords: logic, minitrainer, educational kits

CR Categories: B.1.1 – Microprogrammed logic arrays, B.1.5 – Instruction set interpretation.

1. INTRODUCTION

Teaching the operation of computer hardware has been a feature of courses offered in the Basser Department of Computer Science since its inception.

The design of special purpose equipment for teaching purposes was begun in 1971 by Arthur Sale and Ron Cullen. On the electronic side this featured the use of the then new TTL logic. On the mechanical side they adopted a very flexible system of modular circuit boards on 19inch rack modules. The flexible reconfiguration that this makes possible has been enormously helpful as the Logic Laboratory teaching has evolved.

In 1979 I introduced additional equipment, based on 2901 bit slice chips, to make possible the construction of simple 8 bit CPUs. (Bromley and Nicholson, 1986).

Originally these Computer Architecture labs used a manually loaded ROM and a RAM simulated by the then new Motorola 6800 D2 development kits. For both it was necessary for students to code and load the memory in hex.

More recently the ROM and RAM have been simulated by a PC and programmed by a simple retargetable assembler system, *lass*, implemented first as an awk script. Students are thus able to specify and modify both the computer architecture and its microcode description.

Throughout, the principal target of the Computer Architecture Laboratories has been the design and construction of a conventional Von Neumann single accumulator, three index register machine with a memory of 256 8-bits words. The microprogram is 256 32-bit words, although only about 32 words are typically used. *Lass* supports programming of both the microprogram memory and the target machine memory.

More recently students have implemented RISC architecture machines using the 32-bit memory for program and the 8-bit memory for data. Of course nothing would have prevented us from doing this in 1979 save only that the RISC concept had not then been invented.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

2. LOGIC MINI TRAINERS

In 1997 I embarked on the design of a new series of logic trainers to support a second year course, LL, Languages and Logic, providing a broad base of theoretical background to complement the theoretical content of DDS, Design of Data Structures.

From an engineering perspective the new Logic MiniTrainers marked a move to CMOS from TTL. This made possible a move to battery power which is much more convenient for use in tutorials. Power consumption is about 20mA and four D cells will provide two to three weeks of continuous operation if the equipment is accidentally left switched on.

All CMOS inputs have been tied hi to preserve the TTL convention that all unused inputs take the value of logic 1 – a great convenience in wiring circuits. There are LED indicators on all outputs – another great convenience.

The most obviously unusual feature of the design is the arrangement of the logic gates. This is motivated by the very limited space for connecting pins provided by our standard 19inch mounting hardware and the use of Karnaugh maps, implemented by NAND gates, as a design methodology.

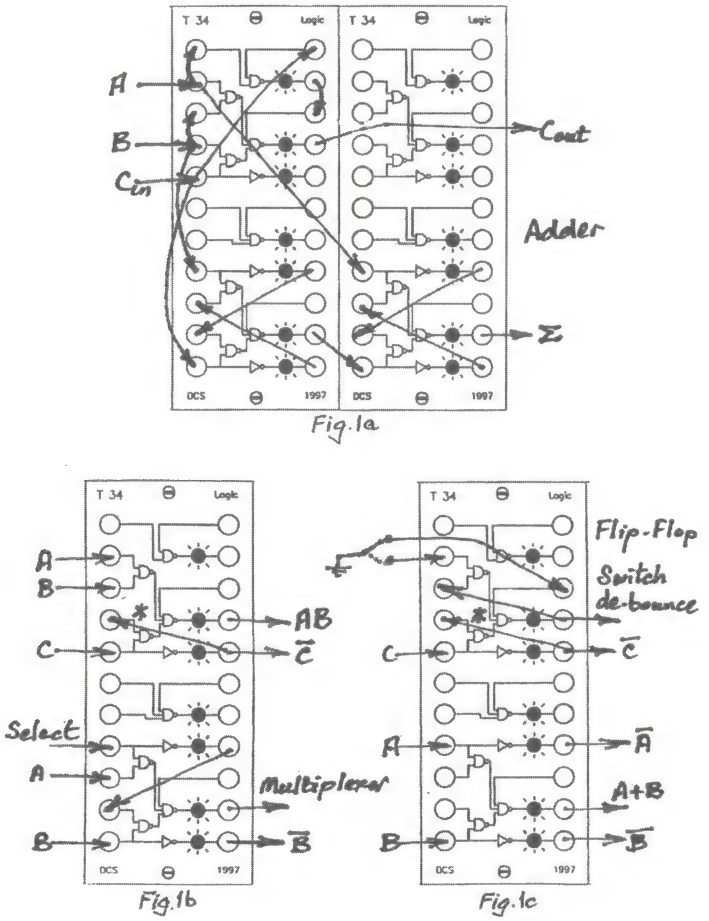


Figure 1: Examples of NAND sum-of products logic

Figure 1 shows some uses of the NAND sum-of-products networks. The sum output of the adder is implemented by exclusive-OR half adders more conveniently than as a direct sum-of-products. The connections in Figure 1b and Figure 1c * disable the lower NAND gate in each network but leave the inverter free for independent use.

Another important design feature is the provision of five D flip flops in one logic panel. Although these were designed initially to optimise the use of connecting pins, I subsequently (re)discovered the great convenience of the Moebius or twisted ring counter. Without this discovery the designs would be much more cumbersome to implement.

E	0	0	0	0	0	1	1	1	1	1
D	0	0	0	0	1	1	1	1	1	0
C	0	0	0	1	1	1	1	1	0	0
B	0	0	1	1	1	1	1	0	0	0
A	0	1	1	1	1	1	0	0	0	0
	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9

The table shows the distinctive pattern of the Moebius counter states. T_0 is used for initialisation, such as setting the carry flip-flop. T_1-T_8 are used for an eight bit serial arithmetic operation. T_9 is used to determine what operation, if any, is to be done during the next clock cycle.

Our clocking convention is that all clock signals are low going pulses active on their rising (trailing) edge. Engineering students often have difficulty because they feel they can design clock circuits without a formal tool like Karnaugh maps.

Eight bit serial arithmetic was chosen as a target of the LL course because it introduces problems of modest complexity whilst emphasising the transition from static logic to the dynamic sequential use of logic in computer design. It has never been the intention to teach how to design hardware – that is the job of engineering teaching – but rather to ensure, in the words of Arthur Sale, that there is "no essential mystery" below the machine language level of programming.

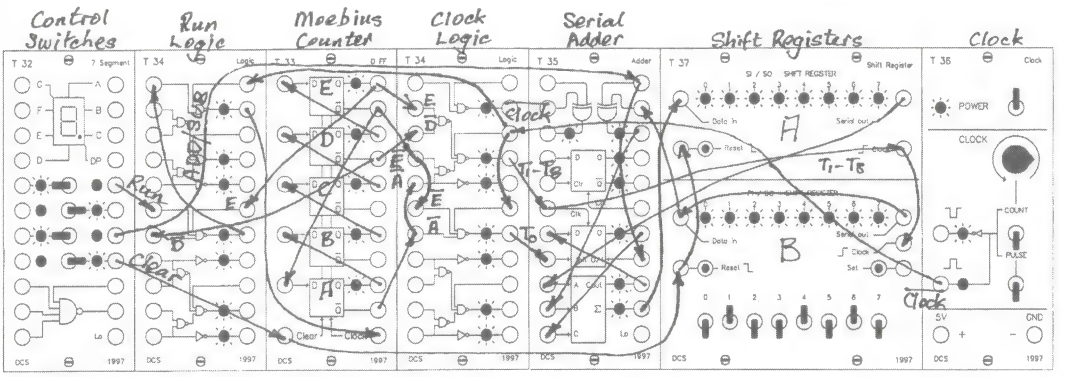


Figure 2: Wiring for the serial adder/subtracter

Figure 2 shows the circuit for serial addition/subtraction. Each operand is entered into register B and taking the Run switch briefly to logic 0 initiates the addition cycle. The clock rate can be varied from 1Hz to 20Hz. Simple modifications to this circuit allows (MOV), ADD, SUB, INC and DEC or the generation of Fibonacci numbers, stopping when the eight bit register capacity is

reached. Another useful experiment is to set flags for negative, zero, carry, and two's complement overflow from arithmetic operations.

Two MiniTrainers may be coupled together to perform multiplication including signed two's complement multiplication using Booth and Wallace recodings.

3. THE FUTURE

I am currently developing the design of the Logic MiniTrainer in kit form suitable for adoption by other institutions and by hobbyists as a learning tool. The parts cost will be around $300. I plan then to revisit the Computer Architecture logic trainers with a view to making them similarly widely available.

REFERENCES

BROMLEY, A.G. and NICHOLSON, T. (1986): Pedagogic Computers. IEE Proceedings. 133 Pt. E (4).

BIOGRAPHICAL NOTES

Allan Bromley has a BSc(Hons) and a PhD from the University of Sydney. His research interests are computer hardware with a particular focus on early calculating devices. He is the world's leading authority on Charles Babbage and has been responsible for cataloguing much of Babbage's work and in conjunction with the Science Museum in London for building working replicas of Babbage designs. He has recently retired from the Basser Department of Computer Science, where he is now an Honorary Associate.

A Sorry Tale – A Study of Women's Participation in IT Higher Education in Australia

Jenny Edwards
Faculty of Information Technology
University of Technology, Sydney
jenny@it.uts.edu.au

Judy Kay
School of Information Technologies
University of Sydney
judy@cs.usyd.edu.au

This paper examines long term changes in the participation of women in professionally accredited computing degree programs. It reports on the results of three intensive Australia-wide studies of the situation in the mid 1980s, in 1992 and in the late 1990s. The early study painted a detailed and rather depressing picture of women's representation in IT education. It also identified barriers to improvement in the discipline itself, the teaching institutions, and for individuals. The intervening years have seen many attempts to address these barriers in respect of both the attraction to and retention of women in IT courses. The current paper summarises the Australia wide studies and then draws upon recent localised data in contrasting environments and larger scale literature to explore how little things have changed as a result of these intervention programs.

Keywords: Computing education, women's participation in IT

CR Categories: K.3.2 – Computer and Information Science Education, K.7 – The Computing Profession

1. INTRODUCTION

There has long been concern at the shortage of IT graduates in Australia (Bennett and Kummerfeld, 1978; Bennett and Kay, 1981). Although there are many ways to address this, one important possibility is to ensure that women are well represented in the IT profession.

The Picture in 1986

In 1986, a CTEC funded study looked at women's representation in professionally accredited computing courses (Kay, *et al*, 1986; 1989). At that stage, it captured a snapshot of courses which had a major role in educating programmers, analysts and other professionals with a high degree of computing skill. It surveyed all departments in Australian CAEs and Universities offering such courses.

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

The study had three major objectives.

- Gaining a detailed quantitative picture of the representation of women in professionally accredited computing courses. In particular, it summarised the number of women starting such courses, the number finishing and the number awarded highest grades.
- Identifying issues that students and staff perceived as relevant for the representation and success of women in such courses.
- Exploring some approaches to supporting women starting such courses.

The study also collected both quantitative and qualitative data to assist with interpreting the statistical data. A smaller number of institutions was selected for student and staff interviews. The study also involved analyses of handbooks and other materials available to students entering the courses.

Major findings on women's representation in the courses in 1984 and 1985 were:

- women constituted about a quarter to a third of the enrolments in most courses;
- at the very early stages, up to the formal enrolment deadline date, women outnumbered men three-to-two among those who discontinued;
- women were slightly less likely than men to pass in the first units;
- there were slightly fewer women at the top of the class in the first units.

However, the study found that in the graduating class of 1985, there were proportionately more women than men at the top of the class.

Overall, the statistical findings indicated problems for women in the early stages of the courses and that some aspects of teaching method, curriculum and course emphasis and orientation seemed to need attention for better outcomes for women in these early stages. Encouragingly, those women who remained, performed well.

Explanations for these findings were explored at three levels:

1. At the disciplinary level, the study observed the commonly projected image of computing to be
 - heavily technology-centred and
 - reliant upon mathematics,

 where those had tended to be less accessible to women and the importance of them for the discipline was frequently overrated. By contrast, other aspects commonly underrated included:
 - the importance of communication and
 - interpersonal skills,

 areas in which women traditionally are believed to have strength.
2. At the institutional level, major issues identified were:
 - effects of assumptions of students' previous computing knowledge and experience;
 - effects of women's typically weaker mathematics background;
 - the coverage and sequence of material within the courses;
 - effects of heavy workload and scarce resources.
3. At the level of the individual, the study identified stereotypes and the attitudes and behaviours that follow from them. Important findings were:
 - Women's participation appeared largely determined either before their entry to tertiary institutions or in the first few months in their courses.
 - Intervention would seem more likely to succeed if it were structured and closely related to the course of study and if it occurred either immediately before or very early in the first year. An example is the orientation program by Kay and Prosser (1989). Similarly for some years, UTS has run a three day (computer-free) orientation camp before the start of the first

semester for all newly enrolled undergraduate computing students. This appears to have enhanced retention rates amongst all students but especially females.

Indeed as most authors of retention program studies have pointed out, the approaches to rethinking curriculum and teaching approaches to make computing education more female friendly, in fact improve it for all students, especially those who might be from other slightly less obvious minority groups.

In all, the study established that women were under represented in IT degrees and there were substantial barriers to a change in this situation.

2. THE PICTURE IN 1992

Cobbin's (1995) study of women's participation in non-traditional fields of study included an analysis of women's enrolment in computer science, information systems and business data processing courses. She detailed women's enrolments in such courses in each Australian university in 1988-92, and provided the representation of women nationally in the sub-fields of study. This suggested some relative growth from 1989 to 1992 in women's commencement and completion of Business Data Processing courses, a relative deterioration in women's commencements and completions in Computer Science courses, and mixed effects in the "Other" and "General" IT&T areas. As an aside, most authors in this field have commented on the disparity of classification of "Computing" courses – a process which makes it difficult to give accurate comparative figures between studies.

3. THE PICTURE IN 1999

Various papers in women in computing and computing education conferences eg Clayton *et al* (1996) and Greenhill *et al* (1996; 1997) have outlined strategies attempted by many University computing departments based on addressing some of the issues raised in earlier studies. More recently, various websites (see references) designed specifically to entice women into the IT profession and give a more accurate picture of the wide variety of careers available have appeared. However, a quantitative and qualitative study looking at under represented groups including women in Australian IT&T carried out for DETYA in 1999 (Cameron, *et al*, 2000) shows that depressingly little has changed in the higher education sector.

Quantitatively, with the exception of scarce resources, the same barriers as those listed in the 1985 study were raised again by staff, students, high school students, teachers and industry. Women now represent some 55% of all Australian university enrolments but some19% of IT&T enrolments. Women in IT&T have an 81% pass rate compared with the overall IT&T success rate of 75%. Slightly over one third of women in IT&T discontinue during or at the end of the first year compared with slightly under one third overall. However, from 1992-1998 the number of women discontinuing in First Year dropped from 37% to 32% indicating that some of the strategies for retention of women might be starting to take effect.

The work of Margolis, Fisher and Miller (2000) and Blum (2001) at Carnegie-Mellon shows that some of their retention strategies such as Big Sister/Little Sister also appear to be having a positive effect. Of course, a significant factor is that US universities unlike most Australian Universities also have the ability to select undergraduate students on attributes other than just academic results. Moreover, Carnegie-Mellon appears to be a shining light with respect to women in computing and their retention experiences are not typical of those being reported in Northern American literature, eg Camp (2001).

3.1 Vocational and Educational Training Sector (mainly TAFE)

Although approximately half of overall VET enrolments in 1998, women were only 17% of VET IT&T course enrolments. Furthermore, they were concentrated in low level entry modules, eg spreadsheets rather than in the higher level Associate Diploma courses. Completion rates are not available but success statistics show women in IT&T at TAFE with success rates of approximately 5% under the overall rate for IT&T at TAFE.

3.2 International

The picture in other English speaking countries is, if anything, even more dismal than in Australia. Camp (1997; 2001) shows the number of females in US Computing courses is steadily decreasing, as is the number at each level from high school to PhD. Similarly in England, the number of females in Computing courses has halved over the last decade. In contrast, however, are Mediterranean countries such as Spain and Italy where the proportion of females in IT courses is around 50%. In countries where IT has strong Government backing, eg Hong Kong and Singapore, the numbers of females are closer to 60% (Edwards, 1998).

In Australia, and particularly in universities in Sydney, an extremely high proportion of local (as well as international) females in computing courses are of Asian, particularly Chinese or Indian, backgrounds. This may be due to existing role models for women in IT in these communities but more likely appears to reflect family views on the suitability of an IT career for women, sutrengthened in part by the emphasis being given IT by many governments in Asian countries. This conclusion is supported as a by product of the work of von Hellens and Nielsen (2001) at Griffith University in Brisbane.

One of the hypotheses in their WinIT study (and others in the literature) was that IT is gender constructed. This was not borne out by their research. The extensive work of Margolis *et al* (2000) at Carnegie-Mellon draws similar conclusions (though based on a smaller sample) on both the numbers of Asian women in Computing and also the fact that IT does not appear to be gender constructed.

4. A CONTRASTING LOCAL STUDY

4.1 The University of Technology, Sydney

On a more local scale – in the mid 1980s at the University of Technology, Sydney (UTS) (then NSWIT), the percentage of women commencing the computing undergraduate course varied from 26% to 44%. On the whole, more men than women discontinued during the first year but the numbers of either discontinuing in or at the end of the first year were small. Graduations were around one third female, ie commensurate with the proportion of females entering. In contrast to many institutions at the time, where Computing was an elective major in say a Science degree, the degree at UTS was essentially all computing. This may explain the low first year discontinuation rate compared with some institutions where students could make the decision each year as to whether or not to continue with Computing subjects. At UTS, students had already made the decision to study Computing for four years and either remained in it or, in rare cases, changed to a different degree.

Despite various initiatives aimed at attracting and retaining more females over many years, the figures from 1992-2000 are little different. Initial enrolments and completions are still around one third female. However, in recent years the proportion of females is much higher (ranging from about 50% to 88%) in the more generalised degrees, the Bachelor of Information Technology, – a

scholarship degree where entry is by both very high academic ability and interview, Computing/Law and Computing/International Studies. Qualitative studies show that students believe these courses offer a broader education with a substantial proportion of non technologically based subjects. Although these courses have comparatively small quotas, unfortunately their success in attracting females has detracted from the proportion of females (now around 27%) in the straight Computing Science degree, which is normally these students' second choice.

Computing is a popular major for students from the Business Faculty. The proportion of females taking the major from the different Business degrees varies but is normally at least 50% for undergraduates, especially amongst international students who are predominantly Asian.

4.2 The University of Sydney

It is interesting to compare the position at UTS with that at University of Sydney as the two institutions represent quite different styles of courses. Where UTS students opt for relatively specialised degree programs, the University of Sydney provides many opportunities for students to undertake broad generalist degree programs. Although there are some quite vocationally oriented programs such as Electrical Engineering (Software Engineering), there are also large enrolments in IT in broader courses. This includes, for example, the Bachelor of Liberal Studies where students can major in computer science but would also study Arts subjects, including languages. Even the Bachelor of Computer Science and Technology allows students considerable freedom and encourages breadth of study. Another new degree in Arts – Informatics, also encourages breadth (and has 44% female enrolment).

The style of courses at the University of Sydney means that students may do varying amounts of IT study and varying flavours of it. So, for example, the foundation programming unit in 2001 has 30% female enrolment, and within that cohort, this varies considerably across faculties. For example among students from the Science Faculty, women constitute 35% of enrolments, in Arts 36%, in Economics 43% and in Engineering just 14%. Perhaps more disappointing is that women make up only 19% of the Advanced class. These figures are consistent with the picture each year since the early 1980s.

Given student choice in their studies, it is relevant to consider the higher year enrolments. The 2001 enrolment in a core second year unit in Computer Science has about 20% women but in third year core units women make up around a quarter of the class. The picture is rather different in the Information Systems units where, for example core and second year units have 33-38% women and by third year women make up almost half of some units.

5. PERENNIAL ISSUES

Studies over some decades, (Kay, *et al*, 1986; Cameron, *et al*, 2000) show that schools continue to play a significant role in the decisions of students, particularly females to study Information Technology. Initially, computing in schools was often associated with mathematics, which was regarded by some females as a disincentive. There were few computers, they were user unfriendly and they tended to be the domain of technically minded males. Computers are now more widespread and somewhat more user friendly. However, in co-educational schools, they are still largely the domain of male students. It is significant that at UTS, the proportion of female first year computing students in recent years from comprehensive co-educational high schools has been as low as 5%. The females choosing computing are overwhelmingly from single sex schools with a few from co-educational selective schools.

Another long standing problem is that many school teachers have still not had the opportunity to become particularly computer literate. A considerable number of school students have access to computers at home and more time to learn about computing than their teachers, though the learning is often at a fairly superficial level. The students' impression is still often that computing consists of spreadsheets, wordprocessing, making webpages and surfing the Internet. They are unaware of the vast body of underlying principles that make up the discipline. In turn, they, and unfortunately their careers advisors, and even the computing teachers have little concept of the wide range of careers available in Information Technology today. Margolis *et al* (2000) used a substantive grant to run workshops for school teachers of advanced computing drawn from across the USA. Apart from benefitting the teachers and all students in their schools, this does appear to have helped increase the proportion of females entering Computing at Carnegie-Mellon (and possibly elsewhere). It was carried out with the benefit of a substantial short term grant. Will the positive effects be long term once the program has finished?

The other significant factor in females rejecting computing as a potential career is that it still has a "nerdy" and often boring image. Despite the efforts of bodies such as the Australian Computer Society (ACS), Australian Information Industries Association (AIIA), Females in Information Technology and Telecommunications (FITT), the National Office for the Information Economy (NOIE) and individual Universities and TAFEs and some IT companies, the general public still thinks of computing in terms of a person (normally a "geeky" male) communing with his computer. The fact that many computing careers consist of working in teams and problem solving with users across a vast range of applications with the added benefits of good pay and often travel is not yet widely understood, especially by school leavers making career choices (interviews in Cameron *et al*, 2000; Margolis, *et al*, 2000). Nor is there an understanding that a computing career is not just technical skills but requires a wider education.

Industry sponsored co-operative programs with scholarships which emphasise these features attract a high proportion of females. While companies are eager to employ these graduates, they are not always as forthcoming with financial support. Governments target undergraduate places in IT. They could contribute in a more practical manner, eg with women in IT scholarships, to ensure that 50% of the population has the opportunity to participate in and contribute to the knowledge revolution.

6. CONCLUSION

The proportion of females in all IT degrees in Australia fluctuates considerably from year to year but there are no apparent trends, except that unlike say the UK or the USA, the proportion in Australia, and certainly at the University of Sydney and UTS is not dropping. Unfortunately, except for the more niche degrees, which tend to have fairly small quotas, the proportion does not seem to be rising either. The overall proportion of females entering IT degrees at these two institutions seems to have reached a state of equilibrium at around one third (lower elsewhere in Australia), despite all the explicit and implicit attempts from individual institutions and industry bodies to attract more. However, the overall proportion of women graduating is now consistent with the entrance proportion suggesting that retention strategies are having some effect, albeit with the women tending to be more concentrated in the 'softer' end of the IT spectrum.

REFERENCES

BENNETT, J. M. and KAY, J. (1981): Quotas and Computer Science. *Proc. ACSC-4 Conference.*

BENNETT, J. M. and KUMMERFELD, J. (1978): Australian Academic Computing and the ASTEC Report. *Proceedings*

Australian Universities Computer Science Seminar, University of New South Wales, Kensington, Australia. Also Basser Department of Computer Science Technical Report No 134.

BLUM, L. (2001): Transforming the culture of computing at Carnegie Mellon. http://www2.cs.cmu.edu/~lblum/women_in_computer_science.pdf.

CAMERON, B., EDWARDS, J., GRANT, J. and KEARNS, P. (2000): Participation in Information Technology & Telecommunications in Education and Training, DETYA Project 99-02. http://www.detya.gov.au/iae/analysis/it_participation.htm.

CAMP, T. (1997): The incredible shrinking pipeline, *Communications of the ACM*, 40(10): 103-110. and http://www.mines.edu/fs_home/tcamp/new-study/new-study.html (2001)

CLAYTON, D., CRANSTON, M. and LYNCH, T. (1996): Attracting and retaining females in information technology courses. *Proc. of the First Australasian Conference on Computer Science Education.* ACM, New York.

COBBIN, D. (1995): Women's participation in nontraditional fields of study at the undergraduate level of higher education 1989-1993. DEET

EDWARDS, J. (1998): Women in computing – Perceptions, realities and visions. Keynote address, *5th Australasian Women in Computing Workshop*, Brisbane.

GREENHILL, A., VON HELLENS, L., NIELSEN , S. and PRINGLE, R. (1996): Asian women in IT education, an Australian examination. *Proc. of the First Australasian Conference on Computer Science Education.* ACM, New York.

GREENHILL, A., VON HELLENS, L., NIELSEN , S. and PRINGLE, R. (1997): Australian women in IT education: multiple meanings and multiculturalism. *In Women, Work and Computerization: Spinning a Web from Past to Future.* GRUNDY, A.F., KOHLER, D., OECHTERING, V. and PETERSEN, U. (eds), Springer-Verlag, Berlin.

GREENHILL, A., VON HELLENS, L., NIELSEN , S. and PRINGLE, R. (1997) Collaborative practice and computer games: towards a new perspective on the participation of women in IT education. *Proc. Fourth Australasian Women in Computing Workshop.* Melbourne.

KAY, J., COLLINGS, P., LUBLIN, J., POINER, G., PROSSER, M., BISHOP. and WATSON, K. (1986): Investigation of the status of women at early stages of professionally accredited tertiary computer science and data processing programs. Report for the CTEC Evaluations and Investigations Program.

KAY, J., LUBLIN, J., POINER, G. and PROSSER, M. (1989): Not even well begun: Women in computing courses. *Higher Education.* Kluwer, 18(4): 511-527.

KAY, J. and PROSSER, M. (1989): An evaluation of a computer science orientation program. *Assessment and Evaluation in Higher Education.* 14(1): 55-66.

MARGOLIS, J., FISHER, A. and MILLER, F. (2000): Various at http://www-2.cs.cmu.edu/~gendergap/papers/.

MCINNIS, C. and JAMES, R. (1995): First year on campus: Diversity in the initial experiences of Australian undergraduates. Australian Government Publishing Service.

VON HELLENS, L. and NIELSEN, S. (2001): Australian women in IT. *Communications of the ACM.* 44(7): 46-52.

http://www.itforwomen.com.au

http://www.it.uts.edu.au/wic

http://www.acs.org.au

http://www.aiia.com.au/careers/careerwomen.html

http://www.aiia.com.au/fitt/mentor.html

http://www.noie.gov.au

BIOGRAPHICAL NOTES

Jenny Edwards gained a BSc, MSc and PhD, all in the Basser Department of Computer Science at the University of Sydney. She was John Bennett's second PhD student. After working at Basser for some years, she joined what is now the Faculty of IT at the University of Technology, Sydney, where she is a Professor and was Head of School of Computing Sciences from 1994-2000. Her research interests include mathematical programming and algorithms for parallel processing. She has also been active in various Women in Computing initiatives and participates in FITT activities.

Judy Kay has a BSc and PhD from the Basser Department of Computer Science at the University of Sydney, where she is now an Associate Professor and a member of the Smart Internet Technology Research Group. Her research interests include User Modelling, User-Adapted Interaction, Scrutable Systems, Artificial Intelligence and Education, ComputerScience Education, Information Filtering, Push Media, Resource Management. She has a particular interest in the exploration of approaches to better teaching and learning of computer science.

Issues in the Deployment of Digital TV on the Internet

Chris Bennett
Vista Systems
San Diego
cjbennett@sbcglobal.net

The Internet has long outgrown its beginnings as a message system. It now hosts a multitude of video and audio applications. However, the current quality of real time delivery is insufficient for streaming media video. This paper examines what is needed to improve this quality to acceptable levels by considering the issues involved for the Internet to support broadcast television.

Keywords and Phrases: streaming media and digital TV on Internet, multicast, quality of service

CR Categories: C.4 – Performance of Systems, H.5.5 – Sound and Music Computing.

1. INTRODUCTION

The last few years have seen rapid growth in "streaming media" applications on the Internet – the ability to view video and audio delivered from a remote site in real time. The variety of existing or planned commercial applications is large, ranging from video conferencing to clip downloads to game playing. Advanced applications such as the Access Grid (AG, 2001), an experimental multimedia framework for distributed group-to-group interaction, are the subject of active current research.

The success of such applications depends on the existence of an infrastructure capable of supporting the delivery of streaming media to a level of quality and reliability that is acceptable to the users. However, for most users of the current Internet the quality of streaming media, particularly of streaming media video, is generally very poor. The pictures are small, the video is jerky and lossy, and servers repeatedly crash when asked to stream a single feed to many users simultaneously. At this point, streaming media services on the Internet remind one of Dr Johnson's dog walking on its hind legs (Boswell, 1791): it is not done well, but the wonder is that it is done at all.

What will it take to make the Internet capable of supporting high quality streaming media applications? Will it be necessary to wait for the deployment of a completely new next-generation infrastructure of the type that is the focus of the Internet2 program? Or can we expect to see significant progress simply through the evolution of technology trends that are already taking place within the current Internet?

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

This paper looks at the issue by considering whether the evolution of the existing Internet infrastructure is likely to be capable of supporting broadcast television. Television is an interesting test case for a couple of reasons. First, unlike, say, the Access Grid, television is an application whose service model is well-understood. Whether delivered over VHF/UHF, cable, or satellite, television is easily the most commercially successful streaming media application that has ever been developed. There can be little doubt that the demand for the service over the Internet would exist if it could be made available. Second, the characteristics of a delivery medium required to support the service – highly-precise real-time, broadcast, high-bandwidth, and essentially unidirectional – are diametrically opposed in almost every respect to the delivery characteristics that the Internet has provided to traditional applications such as electronic mail, file transfer and even web-browsing.

The paper provides a brief overview of the technical issues involved in providing a television service on the Internet. It takes into account evolutionary technology trends that are currently foreseen, rather than quantum leaps such as Internet2 (Teitelbaum, *et al*, 1999). I first review the architecture and the service characteristics of a typical television network. I next give an overview of the prevailing digital television standard MPEG2 and briefly review how it is being integrated into television networks. The architecture and service characteristics of the existing Internet are then compared to the requirements of a television network. Technology trends in upgrading Internet services are then reviewed and assessed for their applicability in allowing television delivery over the Internet.

2. TELEVISION NETWORK ARCHITECTURE AND CHARACTERISTICS

The structure of a typical television network is illustrated in Figure 1. It consists of up to three

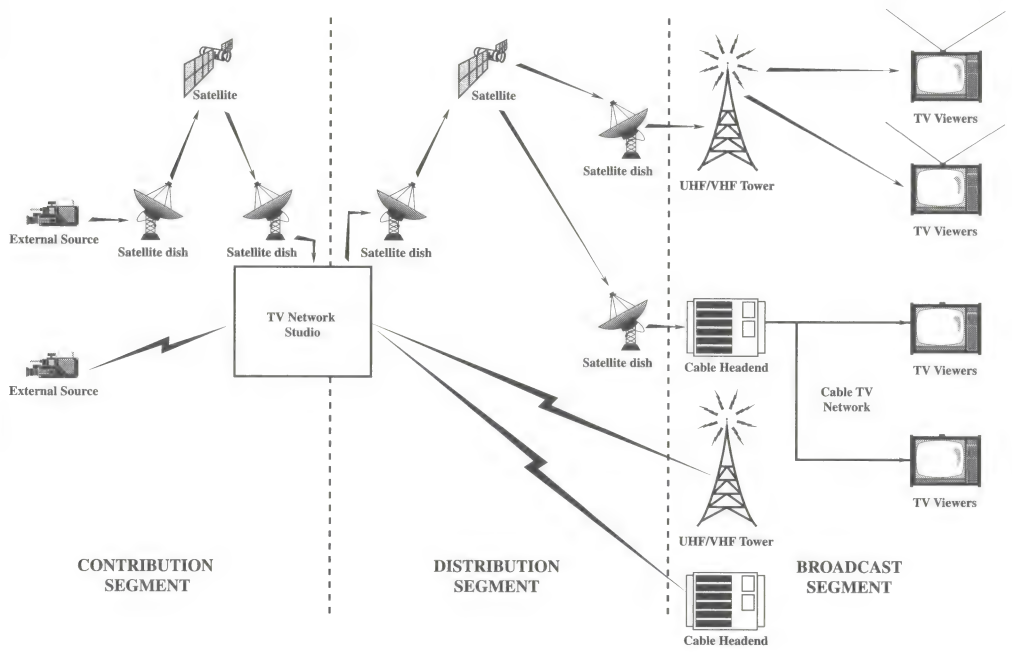


Figure 1: Typical Television Network Architecture

segments or feeds: the contribution feed, the distribution feed, and the broadcast feed. The contribution feed covers the transmission of a television signal into a central processing studio, typically from an outside broadcast unit covering a live event. The distribution feed is a signal which is selected and processed from one of several contribution feeds, or from other sources inside the studio, such as tape or studio cameras. This may be broadcast directly to the consumers, or it may be fed to a set of affiliates, such as local broadcast stations or cable head ends, by a terrestrial transmission system or by satellite. In the latter case, there may be some reprocessing by the affiliates (e.g. for insertion of local advertisements or logos) before the signal is broadcast to the consumer television.

As seen by the consumer, television exhibits the following characteristics:

- It is accessed by a broadcast medium, initially UHF or VHF, more recently by cable or satellite.
- The signal is experienced in real time with a constant delay from the signal source, as continuous video with synchronised audio and possibly with supplementary data signals such as subtitles or teletext.
- The bandwidth available for the signal is fixed by regulatory constraints laid down for historical reasons. Typically, 5.0 or 5.5 MHz per channel is reserved for PAL signals (Europe, Australia etc), and 4.5 MHz per channel for NTSC signals (North America, Japan etc).
- Access is essentially unidirectional. There is very little feedback, if any, from the consumer to the provider in real time. Typically, what user feedback exists is in support of specialised subscription services such as Pay Per View.

3. DIGITAL TELEVISION

Television has been digitised in stages over several decades. Ancillary digital services were added to television signals in the mid 1970s, with teletext and closed caption services. During the early 1980s, digitally-based addressable conditional access systems were deployed, such as the VideoCipher system used to protect access to satellite television in the United States, and the related Eurocypher system developed for use in Europe (Bennett, *et al*, 1990). These systems typically also provided digital stereo audio, primarily because this improved the security of the system by allowing the audio to be digitally encrypted. In Europe at this time, a hybrid analog/digital system called MAC ("Multiplexed Analogue Components" – see (EBU, 1986)) was also fielded that was intended to protect the European electronics industry after the pending expiration of the patents protecting PAL systems. MAC provided multiple digital audio tracks and potential for multilingual subtitling, together with improved video quality. However, despite a large investment by several European governments in MAC systems, the technology was a commercial failure.

The most challenging step in the process has been to digitise the video component. Uncompressed digital representations of the video portion of a television signal require a great deal of information bandwidth: 216 Mbps for a broadcast quality signal, and 1.485 Gbps for an HDTV signal (ITU-R, 1995). Compression techniques permitting digital video signals of acceptable quality to be transmitted in the frequency bandwidth allocated to analogue television were not developed till the early 1990s.

The paradigmatic video standard for digital television is MPEG2 (ISO, 2000b). In an MPEG2 system, each video frame is divided into a sequence of "macroblocks" of 8x8 pixels, which are then subjected to a quantised two-dimensional discrete cosine transform (DCT). Certain frames, known as "intra-coded" frames (or "I-frames") are treated only in this way; the frames between two consecutive I-frames form a Group of Pictures (GOP). Video sequences can only be acquired starting with I frames. The compressed format of other frames in a GOP is formed by comparing

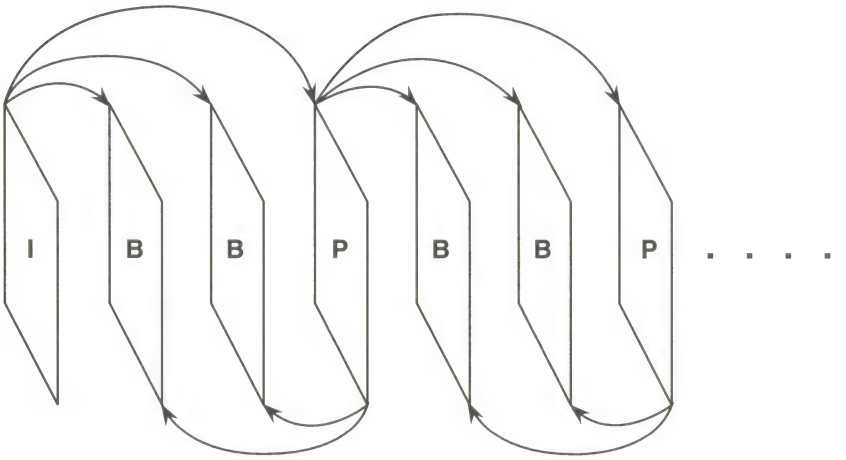


Figure 2: MPEG2 Video Compression – Motion Prediction and GOP Structure

the apparent motion of macroblocks between two frames, and transmitting only the motion vectors applying to those macroblocks wherever possible. Motion estimation can be forward only ("predictive" frames or P-frames), or determined bidirectionally ("bidirectional" or B-frames); this process is illustrated in Figure 2. The latter provides more efficient compression, but in order to allow the frames to be reconstructed sequentially it is necessary to transmit both the frames used for prediction before the B-frame, i.e. frames must be transmitted out of order and buffered for reconstruction. For this reason, MPEG2 allows at most two B-frames between successive I or P-frames. The resultant compressed video stream is then subjected to Huffman coding.

The compression achievable through MPEG2 is significant. First generation professional contribution system products provided contribution-grade quality at 8 Mbps. Consumer-grade signals required about 6 Mbps. As the technology advanced, primarily through the application of advanced encoder techniques (noise reduction prefilters, scene change detection, statistical multiplexing, adaptive preprocessing etc) significant improvements have been made. Typical consumer-grade signals in satellite television systems now require information bandwidth in the 2-4 Mbps range. That is, MPEG2 provides a compression factor of up to 100:1.

In order to transport MPEG2 signals over television networks, the MPEG2 Transport Layer was defined (ISO, 2000a). The Transport Layer divides the set of streams being transported into sequences of 188-byte transport packets, each stream being identified by a separate 13-bit packet ID ("PID"). These individual streams are multiplexed together into a Transport Stream. The key innovation of the Transport Layer is the mechanism used to maintain a timing lock between the transmitters and receivers. The set of PIDs associated with a given service ("program" in MPEG terminology) are also associated with samples of the "program clock reference" (PCR) values for that service, which are 33-bit counts referenced to a 27 MHz clock at the encoder. Decoder synchronisation to this clock is ensured by requiring all decoders to conform to a standard decoder buffer model for recovery of the timing signal from the PCR values. Finally, a directory mechanism is defined that permits the contents of various PIDs to be described, identified and associated through a set of directories transmitted in "tables" in well-known PIDs.

The MPEG2 Transport specifications lay stringent requirements on the underlying network. In particular, the network is required to deliver transport packets with a BER of 10^{-10}, and PCR

timestamps must be delivered with a jitter tolerance of ±500 ns. The mechanisms provided to meet these requirements are not specified by the MPEG-2 standards, but must be defined by supplementary standards. For satellite, cable or UHF/VHF delivery of digital TV, the transmission medium imposes no additional jitter. The BER requirement is met by an appropriate choice of FEC, for example the Digital Video Broadcasting group (DVB) has defined use of a Reed-Solomon (204, 188) code for satellite transmission (ETS, 1994).

Jitter is a more significant issue on multiplexed and switched networks. For example, ATM-based networks are intended to provide switched virtual circuit facilities based on transmission of small 53-byte packets ("cells") that are very suitable for point-to-point transmission of digital video signals. For this purpose the ATM Forum has defined an ATM Adaptation Layer, AAL-1, that provides sufficient error protection to meet MPEG requirements (ITU-T, 1996). However, practical ATM networks have shown PCR jitter due to cell delay variance which is as high as 3 ms. This has led to the development of various proprietary dejittering schemes intended to compensate for these high levels of jitter (Reid and Mulvey, 1995).

4. THE INTERNET AS A TELEVISION NETWORK

The classical Internet is not well adapted for supporting television services. There are three major deficiencies:

- The basic IP internet protocol is designed to provide point-to-point delivery of packets between two communicating parties. Television requires an ability to deliver the packets to multiple destinations.
- The Internet provides no means for assuring the quality of service required for television, of highly reliable and near-jitter-free delivery of television packets. In much of the Internet, jitter far exceeds the worst-case 3ms jitter that is seen in ATM. IP provides no guarantees that packets are delivered uniquely, in order of transmission, or with any limitations on the variability in delay. The transport protocol used to provide reliability of data transmission, TCP, actually increases delay variability by achieving reliability through end-to-end retransmission.
- The classical access interfaces to the Internet – 56 kbps modems, T1 lines (1.544 Mbps) etc. – provide significantly less information bandwidth than the minimum 2-4 Mbps currently required to support consumer-grade digital television.

On the other hand, the Internet automatically provides a return path from the receiver to the transmitter, a feature that cannot be assumed in most television networks. However, this asset must be used carefully, otherwise it will be easy to overwhelm the signal source with feedback from the viewing population.

4.1 Multicast

There are several protocols on the Internet designed to support multicast distribution of data. They all depend on the same basic idea: develop a spanning tree encompassing at least the participants in a multicast group and the IP routers necessary to reach them. The three major protocols are:

- PIM-DM (Protocol-Independent Multicast – Dense Mode). PIM-DM is still being finalised within the Internet Engineering Task Force (IETF). It is based on a "flood-and-prune" approach to building planning trees. All reachable routers are included in building PIM-DM spanning trees. It is intended for effective creation of local conferencing groups, and similar applications.
- PIM-SM (Protocol-Independent Multicast – Sparse Mode). The fundamental difference with PIM-DM is that routers must explicitly request to join a PIM-SM spanning tree. This avoids the

need to maintain spanning tree data in routers which do not actually communicate with members of the group.

- PIM-SSM (Protocol-Independent Multicast – Source Specific Multicast). This is a simplification of PIM-SM currently in development within the IETF. It is designed to build a PIM-SM-like spanning tree based on a single source node. This is a model which exactly represents the situation of distributing television-like signals.

All these protocols are built on top of the underlying Internet routing protocols, and PIM-SM has reached RFC status (Estrin, *et al*, 1998). In practice, however, multicast facilities have not been widely enabled, even when multicast-capable routers have been deployed. The reason is commercial. A well-designed multicast protocol makes efficient use of network resources by deferring the replication of data being delivered to the end user till the last possible moment. However, operators of transit subnetworks within the Internet typically obtain revenue from the total number of packets transferred. Clearly, such operators have no incentive to replace multiple point-to-point transmissions between a pair of routers by a single transmission supporting a point-to-multipoint delivery when the breakout point is downstream from the egress router.

Two developments are occurring which mitigate this by providing countervailing commercial pressures. The first is the deployment of so-called Content Delivery Networks (CDNs) (Stardust.com, 2000). This concept is illustrated in Figure 3. In essence, a CDN is an overlay network based on CDN nodes distributed within the Internet. The CDN nodes are proxy servers with large amounts of storage space that are placed at strategic points by ISPs (Internet Service Provides) affiliated with the CDN operator. Thus, internode transmissions within the CDN appear within the Internet to be separate end-to-end transmissions.

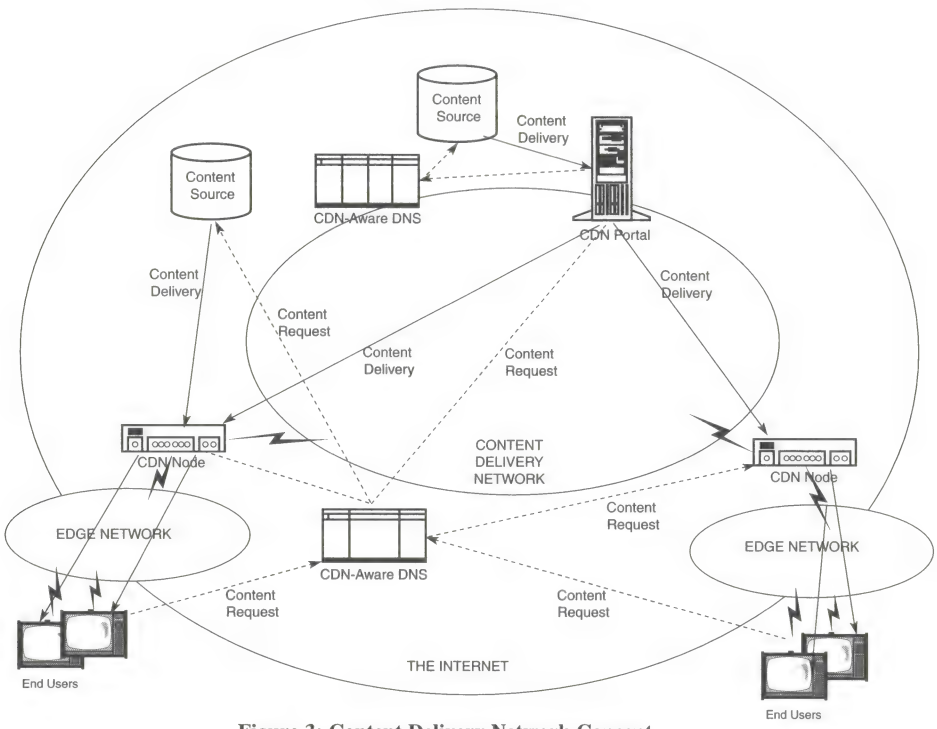


Figure 3: Content Delivery Network Concept

A typical application of a CDN is to replicate widely accessed web material to the CDN nodes, so that end-users are redirected to access the desired material at the nearest local server. URLs requested by the user are resolved by a CDN-aware Domain Name Server (DNS). Typically, the first access to the URL will be made to the actual website, or indirectly through a CDN Portal that accesses the original website, and a copy of the web page is retained at the CDN node. Later accesses can be made by redirecting the user to the copy of the webpage held at the CDN node. Clearly, the same techniques can be used to provide distributed access to streaming video, and television in particular.

For speed and efficiency, the CDN operator may install a private network, so that nodes may be connected through a satellite link, effectively replacing many Internet switches with a single hop. Since the service provided by a CDN is based on distributing multiple copies of widely-accessed material to local CDN nodes, CDN operators have a strong incentive to use multicast techniques within the CDN, in order to minimise their costs of distributing data to the various CDN nodes. This is easily achieved when using satellite links.

Secondly, the edge networks delivering data from the Internet to the end user are becoming increasingly complex, particularly with the emergence of broadband networks based on Digital Subscriber Line (DSL) technology (Goralski, 1998). Since the predominant direction of traffic in such networks is from the network to the end user, and therefore edge networks do not benefit as much as transit networks by charging for packet transit, growth in demand for multicast services such as TV on the Internet is likely to lead to demand for deployment of multicast capabilities within the edge network in order to make efficient use of the available bandwidth. As with satellite-based CDNs, when the edge network itself uses a broadcast medium, such as cable, it is particularly easy to provide a local multicast service.

Thus, in practice, it is unlikely for commercial reasons that multicast protocols will be deployed globally across the Internet by transit subnetwork providers. However, multicast may well be deployed locally in both CDNs and in cable or DSL-based edge networks, because it provides distinct commercial advantage to operators of those networks. Since CDN nodes are typically directly associated with ISPs providing edge network services, the result is to provide a two-stage multicast system. In effect, the problem of multicast is being solved by the creation of an overlay above the Internet that is not unlike the current TV network structure described in section 2, with the CDN playing the role of a distribution network and the edge network corresponding to the final delivery channel.

4.2 Quality of Service

As noted above, the major deficiency in the quality of service provided by the Internet for delivery of television or other streaming media is assuring constancy or near-constancy of delay. Even the jitter of 3 ms seen in ATM networks is far better than the jitter that is seen in the typical Internet environment. However, as with multicast, the problem of jitter is being addressed in the existing Internet by a series of developments that are converging to provide a solution.

Perhaps the least important of these is the development of new classes of routing algorithm that provide differentiated classes of service between packets, or greater fairness in managing network congestion and discard policies, the so-called DiffServ paradigm (Blake, *et al*, 1998). While valuable for many applications, these techniques cannot by themselves address the fundamental causes of the large amount of jitter provided in the classic Internet. These causes are not algorithmic but topological: the number of switches traversed, the existence of alternate routes, and packet loss and discard resulting from the use of alternate routes.

The number of hops in delivery is directly affected by the emergence of CDNs in two ways. First, when CDNs use satellite or direct optical circuits instead of the switched Internet backbone, they directly eliminate the use of switching across CDN links. Secondly, CDN nodes supporting streaming media applications act as edge servers, i.e. they are the local source of the streaming media. In a television application, properly sized edge servers effectively act as very large smoothing buffers for real time data, providing edge networks with the ability to tolerate much larger jitter variations than would be acceptable if the data were provided directly from the data source to the end user.

Jitter which is restricted to edge networks is in general much more manageable than jitter across the wide area Internet. For broadband media such as cable networks, the user is effectively connected to the server by a link that can directly conform to MPEG specifications. Current broadband DSL technology is typically based on delivery across ATM networks, where network induced jitter is a well-understood phenomenon, as discussed above.

When these two features are combined effectively, the result may be that two users associated with different edge servers may not view a television signal supported by such an architecture at precisely the same time. However, each user will experience the illusion of seeing the signal in real time.

A key feature of the reduction of jitter in DSL-based edge networks is the virtual circuit switched nature of an ATM network. The same characteristic is becoming available in optical networks forming the next generation terrestrial backbone for the Internet. Here, fast switching techniques are being deployed based on labels which have been assigned to a route in advance, typically at the time it is created ("Multi-Protocol Label Switching" – MPLS) (Rosen, Viswanathan and Callon, 2001). Since the labels are abstract data constructs, they may be used in principle to implement any desired routing or switching policy. In pure optical systems, MPLS labels may be used to identify different wavelengths – so-called "generalised" MPLS or MPλS (Ashwood-Smith, *et al*, 2001). In practice, the simplest and the first switching policy to be deployed is to create virtual circuits across the optical backbone. In other words, virtual circuit switching is becoming integrated with the per-packet switching techniques native to the Internet. This capability automatically virtually eliminates jitter due to routing.

4.3 Access Interface Capacity

The final technical obstacle to an Internet television service is the bandwidth needed to process the signal. As noted in Section 3, in current consumer-grade MPEG2-based digital television systems 2-4 Mbps is required.

In 1998, the columnist Jakob Nielsen, who considers himself a typical high-end non-commercial user, observed that his connection speed to the Internet had consistently grown at 50% a year since 1984 (Nielsen, 1998). Mr Nielsen's curve predicted that in 2001 he should be able to access the Internet at rates of about 300 kbps, and that by 2004 his access will be at 2 Mbps. Actual available access rates to the Internet in the United States have increased even faster than these anecdotal numbers. Typical ADSL connections can operate today at downstream rates between 800 kbps and 6 Mbps, depending on the length of the local loop. Bandwidth to the home user will increase to 100 Mbps or more with the rollout of FTTH. A similar story is emerging for wireless access. IS-856, known as HDR or 1xEV, currently undergoing field trial, provides mobile wireless access at 2 Mbps (Qualcomm, 2001). The new IEEE 802.11b wireless Ethernet standard provides aggregate data rates of up to 22 Mpbs (Heegard, *et al*, 2001). These numbers are clearly comparable to the rates required

for broadcast-quality MPEG2 television. The practical limitation is not technology but the logistics of commercial deployment.

Only marginal improvements in MPEG2 encoding quality can be expected in future, so it is unlikely that MPEG2-based television will be deployed at significantly lower access speeds. However, the continued application of Moore's Law in processor speeds means that broadcast-quality MPEG2 decoders and even encoders can now be implemented entirely in software running on general-purpose CPUs at reasonable cost. This means that MPEG2 decoders, upgrades to decoders, and if necessary entirely new decoding algorithms can be deployed through software download techniques of the type used today for downloading Web pages.

5. CONCLUSIONS

Television is a very different application from the applications that led to the creation of the Internet. Nevertheless, the technologies now being evolved and fielded in the Internet are creating an environment where we can reasonably expect broadcast quality television to be supportable, at least for high end users within industrially advanced countries, within the next few years. It is not necessary to wait for radical new Internet infrastructure and protocol technologies to emerge.

The service can be supported as a result of evolutionary upgrades that are already being deployed for other commercial reasons. The net effect of these deployments is to create a greatly simplified Internet topology that, for providers of television, is starting to resemble the type of network topology that has been used to supply that service for many years. Content can be broadcast by the provider using CDNs, which inherently support multicast, to CDN nodes, which in turn redistribute the content through edge networks that also support multicast. Jitter is being reduced to acceptable levels by the creation of an overlay delivery topology that eliminates most of the sources of jitter, combined with the provision of sufficiently large smoothing buffers at CDN nodes. Finally, access interface technologies are already reaching the bandwidth levels at which a digital TV signal can be delivered.

Whether Internet TV will ever actually happen is not a technological issue, but a commercial and regulatory one.

REFERENCES

AG (2001): Access Grid. http://www-fp.mcs.anl.gov/fl/accessgrid/default.htm, as of 10 December 2001.

ASHWOOD-SMITH, P., BANERJEE, A., BERGER, L., BERNSTEIN, G., DRAKE, J., FAN, Y., KOMPELLA, K., MANNIE, E., LANG, J. P., RAJAGOPALAN, B., REKHTER, Y., SAHA, D., SHARMA, V., SWALLOW, G. and BO TANG, Z. (2001): Generalized MPLS - Signaling functional description. http://www.ietf.org/internet-drafts/draft-ietf-mpls-generalized-signaling-07.txt, version of November 2001.

BENNETT, C. J., MORONEY, P. and CUTTS, D. (1990): The architecture and security design goals of the Eurocypher system. *Actes des Premières Journées Internationales ACSA '90: Accès conditionnel aux services audiovisuels*, Rennes, France, 61-80.

BLAKE, S., BLACK, D., CARLSON, M., DAVIES, E., WANG, Z. and WEISS, W. (1998): An architecture for differentiated services, RFC 2475, Internet Society, Reston VA.

BOSWELL, J. (1791): The life of Samuel Johnson, LL.D. H. Baldwin, London.

EBU (1986): Specification of the MAC/Packet family. *EBU 3258*, Brussels.

ETS (1994): Digital broadcasting systems for television sound and data services; Framing structure, channel coding and modulation for 11/12 GHz satellite services. *EN 300 421*, Sophia Antipolis, France.

ESTRIN, D., FARINACCI, D., HELMY, A., THALER, D., DEERING, S., HANDLEY, M., JACOBSON, V., LIU, C., SHARMA, P. and WEI, L. (1998): Protocol independent multicast-sparse mode (PIM-SM): Protocol specification, *RFC 2362*, Internet Society, Reston VA.

GORALSKI, W. (1998): ADSL and DSL technologies. McGraw-Hill.

HEEGARD, C., COFFEY, S. T., GUMMADI, S., MURPHY, P. A., PROVENCIO, R., ROSSIN, E. J., SCHRUM, S. and SHOEMAKE, M. B. (2001): High-performance wireless ethernet, *IEEE Communications Magazine* 39(11): 64-73.

ISO (2000a): Information technology – Generic coding of moving pictures and associated audio information: Systems. *ISO/IEC 13818-1:2000*, Geneva, Switzerland.

ISO (2000b): Information technology – Generic coding of moving pictures and associated audio information: Video. *ISO/IEC 13818-2:2000*, Geneva, Switzerland.

ITU-R (1995): Studio encoding parameters of digital television for standard 4:3 and wide screen 16:9 aspect ratios. *ITU-R BT.601-5*, Geneva, Switzerland.

ITU-T (1996): B-ISDN ATM Adaptation Layer (AAL), types 1 and 2 specification. *ITU-T I.363.1:1996*, Geneva, Switzerland.

NIELSEN, J. (1998): Nielsen's law of Internet bandwidth. http://www.useit.com/alertbox/, column of April 5, 1998.

QUALCOMM (2001): 1xEV: 1x Evolution IS-856 TIA/EIA standard. airlink overview, http://www.qualcomm.com/hdr/pdfs/HDR_Tech_Airlink_080900.pdf, Rev 7.1 of May 2001.

REID, A. B. D. and MULVEY, M. (1995): Analytical methods for timing aspects of the transport of CBR services over ATM. *BT Technology Journal* 13(3): 26-34.

ROSEN E., VISWANATHAN A. and CALLON, R. (2001): Multiprotocol label switching architecture, *RFC 3031*, Internet Society, Reston VA.

STARDUST.COM (2000): The ins and outs of content delivery networks: What they are, why we need them and how to build them. http://events.stardust.com/cdnfall/documents/CDN_whitepaper.PDF

TEITELBAUM, B., HARES, S., DUNN, L., NEILSON, R., NARAYAN, V. and REICHMEYER, F. (1999): Internet2 QBone: building a testbed for differentiated services. *IEEE Network* 13(5): 8-16.

BIOGRAPHICAL NOTES

Chris Bennett graduated from Sydney University in 1975 with a BSc. (Hons.) and from the University of London in 1980 with a Ph.D., both in Computer Science. From 1982 to 1991 he worked with M/A-COM Linkabit and General Instrument as Senior Staff Engineer designing secure conditional access systems for satellite television services in Europe and North America. From 1991 to 2001 he worked with Tiernan Communications as Vice President of Systems Engineering and Chief Technology Officer. Tiernan was a leading developer of professional digital television encoders for contribution and high definition television applications. Dr Bennett is currently a consultant with Vista Systems.

Broadband Internet Access in Regional Australia

Arthur Sale
Howrah Tasmania 7018, Australia
ahjs@ozemail.com.au

This article was written to commemorate the Basser Computing Laboratory's early contributions to communications technology by examination of an underused modern technology. It commences with a brief tutorial on spread-spectrum communications, together with a summary of the salient characteristics of the technology. Outdoor and non-mobile uses of Wireless LAN NICs are then introduced with two case studies. In conclusion, the policy implications of spread-spectrum technology for broadband Internet access in regional Australia are examined, together with an analysis of some opportunities for Australian entrepreneurs.

Keywords: broadband services, regional Australia, spread spectrum, Internet access, case study, wireless ISP, national communications policy
CR Categories: C.2.1 – Network Architecture and Design

1. REGIONAL AUSTRALIA AND COMMUNICATIONS

It is little known that Professor John Bennett and the Basser Computing Laboratory were striving to bring computing access to regional Australia even in the late 60s and early 70s. Schools throughout NSW could send pencil-marked (mark-sense) Hollerith card decks in to the University of Sydney, and the programs would be conductively read and compiled. The user would be sent back the compiler listing and any run results. At a rough estimate, a 100-card deck amounted to say 10000 characters each way with a week turnaround, or 100 millibits/sec. This is 100 times smaller than the 10 b/s that was usually achieved as a staff member of Sydney University, but it was valuable.

This paper draws attention to a relatively recent development that is important to communications in regional Australia, and yet which does not seem to be as fully utilised as it might be.

2. SPREAD SPECTRUM WIRELESS NETWORKING

Domestic microwave ovens operate in a narrow band of radio frequencies just above 2.4 GHz. While their emissions are closely controlled through shielding, the sheer number of ovens constitutes a substantial noise source in this part of the radio spectrum. As a consequence, the band is not used by critical commercial services, and most spectrum authorities including the Australian Communications Authority (ACA, 2001) have made 2.400 to 2.484 GHz available for licence-free spread spectrum communications. Similar considerations apply in the band from 5.725 to 5.875

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Guest Editor: Jenny Edwards

GHz. The 2.4 GHz band was enthusiastically adopted by the IT industry for indoor LAN communications (WLANA, 2001). The 5.8 GHz band is just opening up, and only a few off-the-shelf products use it yet.

The free-space wavelength of electromagnetic waves at 2.4 GHz is 125 mm. This implies that a simple half-wave resonant antenna is slightly shorter than 62.5 mm long, and antennas can be quite easily accommodated in PCMCIA cards making this radio spectrum easily available to laptop and desktop computer users without any fuss. At 5.8 GHz these dimensions are halved, making simple antennas even smaller.

None of these communications options would be possible without spread spectrum (SS) technology. The principle of spread spectrum technology is to spread the information to be transmitted over a much wider bandwidth than is necessary, with a characteristic signature that enables it to be picked out from the noise. In most current systems direct-sequence modulation is used (DSSS) whereby each transmitted bit is modulated by a characteristic higher frequency binary sequence. The receiver uses the same sequence to separate this signal from competing signals and genuine noise. By appropriate choice of orthogonal sequences, many different communications can take place in the same channel.

3. WIRELESS LANS

The operation of most digital communications in this band is defined by the IEEE802.11 standard and its successors. The use that was first promoted and which is still the major use of network interface cards (NICs) to IEEE802.11 is local area networking. In fact the radiated power produced by commercial NICs is sized appropriately for this use, in the range 35-50 mW. This provides acceptable indoor range, negligible biohazard, and allows many networks to coexist in the same channel without undue interference.

The basic principle of the infrastructure mode (the most useful to examine) is for one NIC to be connected to a wired backbone LAN. This station is called the *access point*, and acts like a hub. All other stations logged on to the access point use the same channel and spreading sequence, and effectively communicate with the access point in a wireless LAN with total data capacity that may be up to 11 Mb/s. Data rates of 45 Mb/s have been announced.

The NICs are capable of changing their spreading algorithms if the radio channel is particularly noisy or the station signals are very weak, and gracefully degrade to data rates of 5.5, 2 and 1 Mb/s before giving up. The effective range of the LAN from the access point depends on

- the minimum tolerable data rate,
- local electrical noise including competitive WLANs, and
- absorption and reflection of the radio waves by the environment.

About 100 m might be considered normal in built environments. Such WLANs are competitive with wired LANs where the data rates are adequate, at about $A1000 for an access point and $A300 for each connected station. They really shine where one or more of the following apply:

- *Difficult wiring*: retrofitting in heritage buildings, domestic dwellings;
- *Flexible environments*: university and school campuses, convention centres, exhibition centres, in temporary locations for quick set-up and re-use; and
- *Networking mobiles*: wherever laptops congregate, such as airport lounges, hotels, libraries, and increasingly in workplaces.

4. BROADBAND NETWORKING

But this article is not about short-range nor mobile applications, except to note that the volume of these uses has driven the price of the hardware and software down to low levels. Exactly the same NICs can be used for medium range communications up to 40 km, and it is these applications that are so important to regional Australia.

Point to Point

The simplest is the point to point link. This is a do-it-yourself version of the familiar microwave link. There are two NICs, one at each end. To achieve a substantial range three things need to be handled.

- The two points at either end of the path need to have a clear line of sight between them (see also Section 5). It is this clear path that eliminates almost all absorption and most of the reflections.
- The two NICs do not use their in-built antennas, but are connected to directional antennas that direct their power in a torch-like beam to each other, and similarly for reception.
- If extreme range is needed, a bi-directional amplifier may be fitted between the NIC and the antenna. In one direction this boosts the output power to the limits defined by the ACA, and in the other provides a more sensitive and lower noise receiver than the one in the NIC.

Point to point links can be of considerable use in regional Australia by providing broadband access to users located more than a few km from a country town, and in linking multiple access points to achieve coverage of a small country town. The cost of such a link may be of the order of $A2000-4000, and may deliver 11 Mb/s to 10 km, or 1 Mb/s to 40 km.

ISP coverage

Some Australian ISPs provide point-to-point access if requested for 24×7 online customers (for example Southern Internet, 2001). However, the really important application is that of providing wireless coverage over a substantial area at lowest cost. The scheme to be described can drop the per-user cost to around $A700. They are in use in the USA, but rarely as yet in Australia. The concept is to take a WLAN and modify it in some critical areas.

Assume that a high bandwidth Internet link is available in a country town, to which a wireless access point is connected. The access point signal is taken to a nearby high point or mast. An antenna is connected, designed to produce a substantially horizontal radiation pattern, tailored in azimuth to the population area it is designed to serve. A bi-directional amplifier is used, to reach the maximum allowable EIRP (Effective Isotropic Radiated Power).

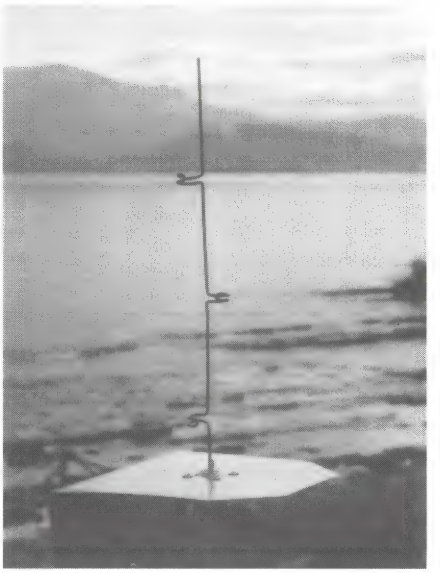
Subscribers to the ISP service must have line of sight to the access point antenna, though community sharing is quite possible. Each subscriber has an NIC and a highly directional antenna pointing towards the access point antenna, again within the ACA limits.

This will achieve a possible 11 Mb/s service area of 5 km radius (assuming an antenna omnidirectional in azimuth), probably capable of supporting 50-70 broadband users with a reasonable fraction of this data rate at most times. Greater range is possible at lower rates, as shown in case study 2. These range extensions have been achieved over the indoor 100 m by eliminating problems in the radio path, providing highly directional antennas, and boosting the central power where highly directional antennas are not possible.

5. THE RADIO PATH AND GROUND SYSTEMS

Outdoor communications require a clear line of sight between sender and receiver at 2.4 GHz and 5.8 GHz. However, this needs to be qualified. At these frequencies attenuation due to water vapour

and rain is negligible (the peak absorption occurs at higher frequencies), and penetration through a few metres of tree foliage may be acceptable. Being able to see the other end is thus not essential at all times. In addition, the e-m waves are also diffracted by large objects. To assure the maximum received signal it is therefore necessary to have a clear zone around the line joining sender and receiver. This includes buildings and the ground. Increasingly at long distances the curvature of the Earth and refraction (bending of the waves) by the atmosphere also play a part in determining received signal strength, and in turn the data rate and reliability of the path.



These bands lie in an awkward region of electronics technology where conventional lumped circuit technology is beginning to break down (a 60 mm wire does not behave like a wire), and microwave technology is still awkward (a copper waveguide with 60 mm side is big). Attenuation in coaxial cables is high, and short runs are essential. Mast-mounting of the radio transmitter and receiver is an advantage.

However, antennas are compact and relatively low cost. Simple omnidirectional antennas can be home-built or purchased, while high-gain antennas up to 24 dBi are based on small parabolic dishes at a cost of say $A200. Limited-gain directional antennas to about 14 dBi are within range of amateur construction at costs of about $A50 (Hecker, 2000). The photograph shows a simple 6 dBi homebuilt collinear antenna (220 mm high) for shore-to-boat communications.

Case Study 1

For the first case study, consider a point-to-point link from an ISP in the City of Hobart and a dwelling on the other side of the River Derwent, 8 km away with the path 90% over water (the photograph does double duty in showing the path). Assume 50 mW is produced by the NICs. Table 1 shows the parameters and design choices.

Terminals	Height	Identifying name	Distance A-B
Point A	13.0m	Howrah (suburb)	8km
Point B	50.0m	ISP connection	90% sea path
Frequency	2.425GHz		
Wavelength	123.6mm		
Desired fade margin	15.0dB	Choose to offset path imperfections	
LOS height at midpoint	31.5m	Check against Fresnel zone clearance	
60% Fresnel zone width	18.9m	Clear zone required around direct LOS	
Onset of d^4 law	> 21.0km	Power drops off faster with distance	

Item	Parameter	Power/Gain/Loss
Radiated power	50mW	**17.0dBm**
Antenna gain Point A		20.0dBi
Antenna gain Point B		20.0dBi
System gains		**40.0dBi**
Free space loss	8km	-118.1dB
Feeder loss Point A	*cable, length*	-2.0dB
Feeder loss Point B	*cable, length*	-2.0dB
Oxygen absorption	8km	-0.1dB
Water absorption	8km	-0.1dB
Allowance for near field tree	3m	**-0.9dB**
System losses		**-123.2dB**
Received power		**-66.2dBm**

Transmission rate	11	5.5	2	**1Mb/s**
Receiver sensitivity at BER = 10^{-5}	-82	-87	-91	-94dBm
Power margin	**15.8**	**20.8**	**24.8**	**27.8dB**

Note:

- The EIRP of each end is calculated by adding the power delivered to the antenna to the antenna gain, so 17-2+20=35 dBm which corresponds to 4 W, the ACA limit between 2.400-2.463 GHz.
- If the noise margin of 15 dB (received power = 30 × noise power) is adequate, communication will be possible at 11 Mb/s.
- On rare occasions the path may be interrupted or degraded for a few minutes by a large passing ship.

Table 1. Link Power Budget, Case Study 1

The conclusion is that most of the Eastern Shore (30% of the population of Greater Hobart) can be serviced with broadband communications *now*, not some time in the future. The number of links is limited so the solution does not scale well, and the installation cost per link is probably around $A2000. However, costs are almost purely incremental with subscribers.

Case Study 2

Consider a small country town on relatively flat terrain except for a single hill of 50 m just outside the limits. From this hill one can see almost all of the houses in the town, and a number of outlying farms. This time put a horizontally omnidirectional antenna on top of the hill on a 10 m mast, with 6 dBi gain enhancing horizontal radiation over vertical. A bi-directional amplifier (Teletronics, 2001) is also fitted, increasing the EIRP to the ACA limit. These steps are designed to give maximum coverage. If the hypothetical hill already has a mobile phone tower or TV tower, it may be possible to locate the ISP antenna on that tower, reducing costs.

Each subscriber to the ISP service (or group of subscribers; there is nothing to stop a group of people in a block of units or a shopping mall pooling a radio link with some bandwidth competition) will install a directional antenna pointing directly at the hill, and use a standard NIC. A more

conservative fade margin of 20 dB (100 × noise power) will be used to recognize the more cluttered environment.

Either direction may be the critical one. At each point the maximum EIRP is radiated towards the other point, so the differences lie in the receiving antenna gains and receiver sensitivities. Table 2 shows the maximum distance at which 2 Mb/s service is possible (15 km), for approximate comparison with a top-level Telstra ADSL service (up to 1.5 Mb/s download and up to 256 kb/s upload).

Terminals	**Height**	**Identifying name**	**Distance A-B**
Point A	60.0 m	Lookout Hill	15 km
Point B	5.0 m	Subscriber	
Frequency	2.425 GHz		
Wavelength	123.6 mm		
Desired fade margin	20.0 dB	Choose to offset path imperfections	
LOS height at midpoint	32.5 m	Check against Fresnel zone clearance	
60% Fresnel zone width	25.8 m	Clear zone required around direct LOS	
Onset of d^4 law	> 9.7 km	Power drops off faster with distance	

	Parameter	**Power/Gain/Loss**	**Power/Gain/Loss**
Radiated power Point A	1000mW	**30.0 dBm**	
Radiated power Point B	50mW		**17.0dBm**
Antenna gain Point A		6.0dBi	as at left
Antenna gain Point B		20.0dBi	as at left
System gains		**26.0dBi**	**26.0dBi**
Free space loss	15km	-123.5dB	as at left
Feeder loss Point A	*cable, length*	-1.0dB	as at left
Feeder loss Point B	*cable, length*	-2.0dB	as at left
Oxygen absorption	15km	-0.1dB	as at left
Water absorption	15km	-0.2dB	as at left
System losses		**-126.8dB**	**-126.8dBi**
Received power		**-70.8dBm**	**-83.8dBm**

Transmission rate A->B	11	5.5	2	**1Mb/s**
Receiver sensitivity at BER = 10-5	-82	-87	-91	-94dBm
Power margin	11.2	16.2	**20.2**	**23.2dB**
Transmission rate B->A	11	5.5	2	**1Mb/s**
Receiver sensitivity at BER = 10-5	-96	-101	-105	-108dBm
Power margin	12.2	17.2	**21.2**	**24.2dB**

Table 2. Link Power Budget, Case Study 2

This analysis must be modified if the surrounding terrain is hilly so the coverage is worse, or if the town has a ribbon configuration so that parts of it are out of range (such as along the Mornington Peninsula). A totally flat terrain can also cause problems in getting the station antenna high enough. However, multi-station configurations are easily possible, and indeed would be essential to fully service a regional city of say 20000 people. It is estimated that the wireless links of a single station should cope with the broadband access needs of say 50-70 domestic and small business subscribers, and central station costs would be split amongst them. Stations can also be co-located in dense usage areas with sectoral coverage.

Price comparison

It is difficult to provide directly comparable rates, but rough analyses suggest that wireless ISPs should be able to provide broadband access to the Internet in towns and cities in regional Australia for significantly less than the price of ADSL links as published by Telstra (Xnet, 2000).

Urban environment

Outdoor use of the 2.4 GHz spectrum in densely populated urban areas such as Sydney and Melbourne is less important, because of limited scalability, interference, multi-path cancellation due to reflection from buildings and vehicles, and a higher electrical noise environment. Wired solutions, whether copper-based or optical fibres, will continue to be more important. There are some applications however for point to point links between buildings.

6. SECURITY

Wireless communications are intrinsically subject to eavesdropping; security and unauthorised access are serious concerns. Suitable deployment techniques have to be used to ensure security is maintained. There is insufficient room in this paper to present an in-depth analysis of the issues, but the most important features are:

- IEEE802.11 communications use Wireless Equivalent Privacy (WEP), which is based on the RC4 encryption algorithm and its huge state space. RC4 is considered a secure algorithm.
- Encryption is optional and some installers do not turn it on.
- Brute force cracking of the WEP 40-bit keys is feasible in months with PC arrays or supercomputers; however many systems use 128-bit keys.
- Recently, Fluhrer, Mantin and Shamir (2001); Stubblefield, Ioannidis and Rubin (2001); Loeb (2001) demonstrated that a particular weakness in TCP/IP traffic could be exploited in some situations to crack WEP keys in minutes to hours on a PC. The time is only weakly dependent on key length.
- In many broadcast systems all users have the same key, which presents insecurity through human leaks.

IEEE80-2.11 communications *must* be secured at the network level, for example by techniques like the Secure Socket Layer (SSL). The wireless access points should communicate with the wired Internet through a firewall, as a virtual private network. Appropriate authentication is also needed.

7. POLICY

At present, Australia is allocating substantial funds to make Internet access available to regional Australia, with broadband access being a target. The policies are mainly driving along two paths:

satellite access and ADSL. Satellite access can be discounted as an option for all but the most difficult and remote users, for the following reasons.

A ground station to link to a geostationary satellite is a high initial cost, increasingly so in southern Australia as large dishes are required to view the satellite at low elevations. For example, in southern Tasmania satellite broadband must be considered to be only just feasible for domestic consumers.

The service is asymmetrical: the high speed is from the Internet to the user via the satellite, with upload being through a modem. This does not help when the user is engaged in an electronic information exchange with another user, such as sending large attachments (pictures, designs, documents, software) with email. Delivery of on-line services or mounting a web-server is difficult. Even uploading web site changes to an urban ISP is a chore. Webcams and online control via video are impossible. All this limits the development of IT industries and usage in regional Australia.

ADSL provides a feasibly costed service, though the less marked asymmetry is still a problem. The key problems are the cost compared to wireless, the limited speed, and the painfully slow rollout of ADSL to regional Australia. For the highest per capita income suburb in Tasmania (Howrah/Tranmere) potential subscribers are told 'We have no current plans to provide ADSL to your area' (Telstra, 2001). However, ADSL services are good for Telco shareholders, because they enhance the value of the monopoly on copper wire 'last-mile' connections.

In short, satellite and ADSL are desirable services, but they are mostly relevant to really remote users and dense urban environments respectively. Country towns and regional cites can be provided with broadband services at lower cost almost immediately, through the use of spread spectrum technology. Why has this not happened? Probably four related reasons are important:

- 'Big-picture' solutions (satellite, ADSL) fit better with the thinking of politicians and Telcos.
- Spread spectrum is a grass-roots technology; any small ISP can set up a wireless broadband service. Indeed so can amateur groups. This discourages Telcos.
- ADSL is to Telstra's advantage in maintaining market dominance and preselection through leverage of the infrastructure monopoly.
- Spread spectrum technology has an image of 'short-range WLAN' in Australia.

Take-up of ADSL in a *Networking the Nation* trial in Launceston (northern Tasmania) resulted in only 180 subscribers (target of 2500 by mid 2002). The Commonwealth Government recently announced a subsidy to waive the $A189 connection fee and reduce the monthly fee to $A50.50 from $A89 (*Mercury*, 11-07-2001). It would be interesting to see the effect of an equivalent subsidy ($A1.1M over two years) applied to wireless broadband in a similar city without broadband services.

It may also be noted that mobile phones are being adopted in developing nations rather than upgrading the poor wired infrastructure. In many countries the number of mobile phones exceeds fixed phones. The situation with the Internet in regional Australia has many parallels with this scenario.

In the USA, high speed access to 2 Mb/s via CDMA phone technology (also spread spectrum) has been developed (Qualcomm, 2001). However, this technology appears to be more suited to occasional high download bandwidth for mobile users, rather than the 24×7 and high-speed upload access required for web-servers and e-commerce (Bender, Black, Grob, Padovani, Sindushayana and Viterbi, 2000).

8. OPPORTUNITIES AUSTRALIA MAY BE MISSING

There are entrepreneurial opportunities for several groups from spread spectrum technology.

- *Antenna design.* This niche is probably filled by Pacific Satellite (2001), but with aggressive increase of Australian expertise in SS, the demand for expertise in antennas and connection equipment could grow substantially, internally as well as for export in the Australasian region.
- *Migration to 5.8 GHz.* Spread spectrum NICs operating in this band are few, though there is an upconverter that translates 2.4 GHz signals to 5.8 GHz and vice versa, at extra cost. Integration of antennas and systems is still some way off. There may be scope for engineering development in Australia focused both on 5.8 GHz and ruggedized outdoor SS systems, provided concepts are rapidly turned into product and export markets are aggressively sought.
- *Consulting.* SS links are low cost and grass roots solutions. At present, most systems are set up in conjunction with supplier advice. A common experience is that this tends to be conservative and more expensive than necessary. There may be room for consultancy services specialising in SS.
- *ISPs.* Adding a wireless 'last mile' strategy may benefit many ISPs and improve competition in the provision of broadband services, to the consumers' benefit.
- *Education.* Graduates intending to work in the network side of the IT industry must have the basic rudiments of radio and SS technology, for the outlined reasons and for mobile communications generally. The value of engineering and computing degrees to international students could be enhanced, with export consequences.
- *Roll-out.* The policies of the Commonwealth and of the political parties should provide incentives (parallel to the ADSL and other initiatives) for the establishment of wireless ISPs in regional Australia, as soon as possible. Community access centres should also be capable of managing such funds and services.

REFERENCES

AUSTRALIAN COMMUNICATIONS AUTHORITY (2001): *Radiocommunications class licence (Spread Spectrum Devices).* Available at http://www.aca.gov.au/publications/info/spreadsp.htm.

BENDER, B., BLACK, P., GROB, M., PADOVANI, R., SINDUSHAYANA, N. and VITERBI, A. (2000): *CDMA/HDR: A bandwidth efficient high speed data service for Nomadic users.* Available at http://www.qualcomm.com /hdr/pdfs/CDMA_HDR_IEEE.pdf.

FLUHRER, S., MANTIN, I. and SHAMIR, A. (2001): *Weaknesses in the key scheduling algorithm of RC4, J.Cryptology.* Available at http://www.simovits.com/archive/rc4_ksaproc.pdf.

HECKER, J. (2000): *How to make a simple 2.425GHz helical aerial for wireless ISM band devices.* Available at http://users.bigpond.net.au/jhecker/.

LOEB, L. (2001): *What's up with WEP?* Available at http://www-106.ibm.com/developerworks/library/s-wep/?dwzone=wireless.

McLARNON, B. (2001): *Wireless LAN/MAN product directory.* Available at http://hydra.carleton.ca/info/.

PACIFIC SATELLITE (2001): *Home page, antenna systems.* Available at http://www.pacsat.com.au/.

QUALCOMM (2001): *HDR Home page.* Available at http://www.qualcomm.com/hdr/.

SOUTHERN INTERNET (2001): *Permanent high speed services.* Available at http://www.southcom.com.au /services/24x7fast.html.

STUBBLEFIELD, A., IOANNIDIS, J. and RUBIN, A. (2001): *Using the Fluhrer, Mantin, and Shamir attack to break WEP.* Available at http://www.cs.rice.edu/~astubble/wep/.

TELETRONICS (2001): *Bi-directional amplifiers and 5.8 GHz upconverter.* Available at http://www.teletronics.com/.

WLANA (2000): *WLAN Association.* Available at http://www.wlana.org/

XNET (2001): *Mornington Peninsula wireless network.* Available at http://www.x.net.au/mornington_server.html and http://www.x.net.au/wireless_peninsula.jpg.

ZDNET (2001): *Summary of features.* Available at http://www.zdnet.com/products/stories/reviews/0,4161, 2684722,00.html.

BIOGRAPHICAL NOTES

Arthur Sale FACS FIEAust gained his PhD in Binary Arithmetic Algorithms. He worked at Philips NV (Netherlands) and the University of Natal (South Africa) before joining the Basser Computing Laboratory in 1969. After active involvement in five years of change, he left Sydney to take up a Chair at the University of Tasmania to establish its computer science department. He helped establish the Tasmanian Branch of the ACS, chaired its Branch Executive and served as National Vice-President of the ACS. He was Chair of the University's Professorial Board, and later Pro Vice-Chancellor (Information Services) with two Australian awards for client service quality to his Division's credit. In 1999 he left the University and now undertakes consultancies in mobile and ubiquitous computing, teaches a few graduate students for fun, and makes glass art some of which can be seen on his website http://arthur.sale.tripod.com/.

B-Trees – A Simple Proof of Their Space Utilisation

J. Hext

Department of Computing, Macquarie University
Sydney, NSW 2109, Australia
jan@ics.mq.edu.au

It is well known that under random insertions the space utilisation of B-trees is approximately ln 2. This paper provides a simple proof.

Keywords: B-Tree, space utilisation.
CR Categories: D.2.8 – Performance measures, E.1 – Trees.

1. INTRODUCTION

Yao (1978) showed that under certain conditions the expected storage utilisation of a B-tree is ln2 – that is, approximately 69%. This means that nearly one third of the space allocated to it will probably be unused. Since B-trees, in one form or another, are the most common data structure for storing large files of records, the result has great practical importance. Its simplicity also raises the question: should we somehow intuitively *expect* it to be ln2? If seen in the right way, should this be "obvious"? Does it have a simple proof?

Since Yao's proof was far from simple, Leung (1984) offered an alternative. He remarked that "it is sometimes not easy to see from the formidable mathematical analysis why *intuitively* this neat approximation should be true." Unfortunately, though, his proof was based on a false assumption. This was pointed out by Gupta and Srinivasan (1986), who outlined a way of making his proof more rigorous. However, the resulting analysis was as formidable as Yao's and so they did not carry it through.

Since then, the ln 2 result has often been quoted but a proof is seldom, if ever, given. The purpose of this paper is to offer a simple proof. Whether or not it makes the result "intuitive" is for the reader to judge.

2. DEFINITIONS AND ASSUMPTIONS

The analysis assumes that the B-tree represents a file of N fixed-length records. The records are stored in ascending key sequence across a series of logical blocks (or "nodes"), each capable of holding up to $2m - 1$ records for some integer m. If there are B such blocks, their storage capacity M is $(2m - 1)\ B$ records and their *space utilisation factor*, or SUF, is defined to be N/M. This definition assumes that a full block, containing $2m - 1$ records, has no surplus space and that the space required for indexing is negligible.

When a record with a new key is added, it is inserted at the appropriate point in the sequence. If that happens to be in a block that is full, the block is *split*: a new block is created and the $2m$ records are divided equally between the old one and the new one. This is the defining characteristic of B-trees. When a record is deleted, an inverse process occurs, possibly resulting in two adjacent blocks

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

being merged. However, the 1n 2 result does not allow for deletions. It applies only to random insertions. (If deletions are allowed, the SUF will almost certainly be worse.)

The N keys, ending with $+\infty$, define N intervals into which the new record may be slotted. If it has key k, it is inserted between the records with successive keys k' and k'' such that $k' < k < k''$. In practice, this usually means that it is inserted in the block containing k''. The question arises: if k is chosen randomly, what is the probability that it will belong to any particular interval? Since no prior assumptions are made about the distribution of the keys, the only way to answer the question is on the basis of their *current* distribution. Noting that the existing keys are distributed one per interval, we therefore proceed on the basis that *all intervals are equally likely*. This is known as the "random insertion assumption". An immediate consequence is that the probability of a new record landing in a particular *block* is proportional to the number of records already in it. Informally, this means that the popular blocks continue to be popular.

3. A SIMPLE PROOF

Let us define an *i-block* to be a block with i records and b_i to be the number of i-blocks. It follows that the number of blocks B is given by:

$$B = \sum_{i=m}^{2m-1} b_i \tag{1}$$

and that their total capacity M is given by:

$$M = (2m-1) \sum_{i=m}^{2m-1} b_i \tag{2}$$

Similarly, the number of records N is given by the formula:

$$N = \sum_{i=m}^{2m-1} ib_i \tag{3}$$

These two formulae will be used to calculate the SUF which, as we noted earlier, is defined by:

$$SUF = N/M \tag{4}$$

Next, let us define $p_i = ib_i/N$ to be the proportion of records that are in i-blocks. We assume that under repeated insertions, but no deletions, the expected proportions p_i remain constant.

Now consider the effect that a random insertion has on the expected number of i-blocks, b_i. Since the expected proportions p_i remain constant, the expected value of b_i increases from Np_i/i to $(N+1)p_i/i$. How is the increase of p_i/i to be accounted for?

Provided that $i > m$, there are only two possibilities to consider. The first is that the new record is added to an $(i-1)$-block, causing b_i to *increase* by 1. The second is that it is added to an i-block, causing b_i to *decrease* by 1. According to the random insertion assumption, as discussed above, the first happens with probability p_{i-1} and the second with probability p_i. It follows that for i in $m+1..2m-1$:

$$p_i/i = p_{i-1} - p_i \tag{5}$$

This gives the recurrence relation:

$$p_i = \frac{i}{i+1} p_{i-1} \tag{6}$$

which has the simple solution:

$$p_i = \frac{m+1}{i+1} p_m \tag{7}$$

In other words, p_i is proportional to $1/(i+1)$. It follows that for i in m .. $2m-1$:

$$b_i = \alpha / i(i+1) \tag{8}$$

for some constant α.

This simple property provides the essence of the proof. It may seem surprising that no reference has been made to the splitting of nodes when a random insertion is made to a $(2m-1)$-block. However, this "wraparound" effect is inherent in Equation (7) for the case $i = 2m-1$. It yields the relation:

$$p_m / m = 2p_{m-1} - p_m$$

which is the equivalent of (5) for m-blocks.

To complete the proof, we use (8) to substitute for b_i in (3) and (2). Substituting in (3) gives:

$$\begin{aligned} N &= \sum_{i=m}^{2m-1} \alpha/(i+1) \\ &= \alpha(H_{2m} - H_m) \end{aligned}$$

where H_n is the n^{th} harmonic number, defined by:

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

For large n, H_n is approximately $\gamma + \ln n$, where γ is Euler's constant, namely 0.577... . So N is α ($\ln 2m - \ln m$), which is simply α 1n 2.
Substituting in (2) gives:

$$\begin{aligned} M &= (2m-1) \sum_{i=m}^{2m-1} \alpha/i(i+1) \\ &= (2m-1)\alpha \sum_{i=m}^{2m-1} \left(\frac{1}{i} - \frac{1}{i+1}\right) \\ &= \alpha \frac{2m-1}{2m} \end{aligned}$$

Finally, substituting in (4) gives:

$$\begin{aligned} SUF &= N/M \\ &= \frac{2m}{2m-1} \ln 2 \end{aligned}$$

which, for large m, is approximately 1n 2.

4. MAKING IT SIMPLER

The above algebra is still not as simple as we might like. So, as an alternative, let us extrapolate from the discrete case to the continuous case and apply some basic statistics.

Each block has its own SUF lying in the range 0.5 to 1.0. For an i-block, it is $i/(2m-1)$. Let $f(x)$ be the probability density function for blocks whose SUF is x. Then by definition:

$$\int_{0.5}^{1.0} f(x)dx = 1 \tag{9}$$

Furthermore, the average SUF across all blocks is:

$$\int_{0.5}^{1.0} xf(x)dx \tag{10}$$

To determine $f(x)$, we return to Equation (8), which states that in the discrete case the expected number of i-blocks is proportional to $1/i(i+1)$. Putting $i = (2m\text{-}1)x$ shows that it is therefore proportional to $1/x(x+\varepsilon)$, where x is their SUF and $\varepsilon = 1/(2m\text{-}1)$. Since $x > 0.5$, we can deduce that, in the continuous case, $f(x)$ is proportional to $1/x^2$. Substituting in (9) shows that it is exactly $1/x^2$, and that $x\,f(x)$ is therefore $1/x$. From (10) it follows that the overall SUF is 1n 2.

5. INTUITION

The key to the proof lies in Equation (5), from which (8) readily follows. From there, the algebraic approach provides an exact analysis for the discrete case, leading to the 1n 2 result in the limit. The statistical approach bypasses the discrete case and leads to the 1n 2 result more quickly.

In terms of our original objective, this completes the quest for a simpler proof. But does it make the result any more "obvious" or "intuitive"? To push further towards that goal, it may help to focus on the fact that the number of m-blocks must be approximately *four times* the number of $(2m\text{-}1)$ – blocks. This follows from a consideration of (a) the splitting process and (b) the random insertion principle. Once we have seen why it should be so, we might plausibly guess that the probability density function $f(x)$ is proportional to $1/x^2$. From there, as we have noted, the result readily follows.

If informed guesswork can be described as intuition, this brings us close to the goal. On the other hand, people versed in mathematical and numerical analysis may yet find a better route to the result, using techniques that avoid the guesswork entirely. If so, we await their contributions with interest.

6. REFERENCES

GUPTA, G.K. and SRINIVASAN, B. (1986): Approximate storage utilisation of B-trees. *Information Processing Letters*, 22(5): 243-246.

LEUNG, C.H.C. (1984): Approximate storage utilisation of B-trees: a simple derivation and generalisation. *Information Processing Letters* 19(4): 199-201.

YAO, A.C-C. (1978): On random 2-3 trees. *Acta Informatica* 9(2): 159-170.

BIOGRAPHICAL NOTES

After gaining a BA in Mathematics and a PhD in Computer Science from Cambridge University, Jan Hext moved to Australia and to a lectureship at the University of Sydney. In 1981 he was appointed to the new Chair of Computing at Macquarie University, where he is now an Emeritus Professor. His main interests are in computer software, particularly programming languages, data structures, algorithms and real-time systems. He is a Member of the ACM and the IEEE, and a Fellow of the ACS.

Apparent Nondeterminism in Actions

Norman Foo

School of Computer Science and Engineering,
University of New South Wales, Sydney NSW 2052, Australia
norman@cse.unsw.edu.au.

Rex Kwok

School of Computer Science and Engineering,
University of New South Wales, Sydney NSW 2052, Australia
rkwok@cse.unsw.edu.au.

Yan Zhang

Department of Computing, University of Western Sydney,
Nepean, Kingswood, NSW 2747, Australia
yan@cit.nepean.uws.edu.au.

Nondeterminism has a central role in computer science, and poses difficulties for many formalisations of domain knowledge in artificial intelligence. In particular, the nondeterminism of actions often complicate the formal treatments of actions. This paper elucidates some possible sources of nondeterminism in theories of actions. The emphasis is on how nondeterminism may arise from inadequacies of the representation language, the identification of which can suggest strategies for reducing or eliminating the apparent nondeterminism.

Keywords: nondeterminism, actions
CR Categories: F.1.2 – Alternation and nondeterminism

1. INTRODUCTION

Nondeterminism occupies a central role in theoretical computer science, perhaps most strikingly as the *P=NP* question. The issue there is whether the apparent power of nondeterminism is "real", for if it is demonstrably not the case, we would have efficient algorithms for ostensibly hard problems. It is less well-known that nondeterminism also poses difficulties and challenges in areas where computational efficiency may not be the central issue. This paper focuses on one aspect of this problem, viz, that in which the nondeterminism is apparent only because the domain representation has omitted essential information. It should not be surprising that if some aspects of "reality" are not captured by a formal model, it may prove to be deficient. On the other hand it is impossible for a model to be entirely faithful. The issue is to decide what to omit. Three common omissions that lead to avoidable nondeterminism are addressed in Section 4, viz, that which assumes too simple a

Copyright© 2001, Australian Computer Society Inc. General permission to republish, but not for profit, all or part of this material is granted, provided that the JRPIT copyright notice is given and that reference is made to the publication, to its date of issue, and to the fact that reprinting privileges were granted by permission of the Australian Computer Society Inc.

Manuscript received: July 2001
Associate Editor: Jenny Edwards

version of the Markov property, that which fails to represent the direction of causation, and that which has insufficient descriptive granularity.

2. A FORMALISM FOR DYNAMIC DOMAINS

Our main concern will be with dynamic domains. These domains evolve with time due to actions performed on their objects. In order to focus the discussion, we will confine attention to discrete-time domains in which time is modelled by the natural numbers. Moreover, all objects in the domain are discrete and have distinct names; they also have have properties that can be expressed as propositions. Actions on the domain at time n will have effects at time $n+1$. Informally, we want to be able to say things like: "After toggling the switch the light will come on"; "If the block is not too heavy, the robot arm can lift it"; and "Blocking a vent will cause the room to be stuffy". We may also wish to say things which are indefinite, like: "If we drop a block on a checker-board, it will land on a white or a black square". Fortunately, there is a ready-made formalism to do this – the Situation Calculus (SC) (Shanahan, 1997). The SC is a multi-sorted first-order language, often augmented with second-order axioms to select preferred models. The sorts are *Act* for actions, *Flu* for fluents, and *Sit* for situations. Fluents are propositions that express properties of the objects of interest, e.g. *switch-on or stuffy*. Situations are configurations of the world as results of actions (unless the dynamic system has the Markov property (see Section 4), the history of the action sequence may matter, and not just the configuration of fluents). There is a binary function *Result* : $Act \times Sit \rightarrow Sit$, and a binary predicate *Holds* : $Flu \times Sit$. A distinguished member of *Sit* is the initial situation $s0$ (the "Garden-of-Eden situation") from which all situations originate. Hence, the situation that results from applying the actions a and b in succession is denoted by *Result(b,Result(a,s0))*. The fact that a fluent f holds (respectively, does not hold) in this situation will be written as *Holds(f,Result(b,Result(a,s0))* (respectively, $\neg$ *Holds(f,Result(b,Result(a,s0))* . $\neg$ *Holds(f,s)* is alternatively written as *Holds($\neg$ f,s)* , it being understood that the latter is an abbreviation for the former. A *fluent literal* is either a fluent f or its negation $\neg f$. The set of fluent literals is denoted by *Lit*.

Here is an example of the familiar blocks world partial specification using the SC.

$$\begin{aligned} &Holds(On(A,B),Result(Stack(A,B),s)) \wedge \\ &\quad \neg Holds\ (Clear(B),Result(Stack(A,B),s)) \\ &\quad \leftarrow [Holds(InHand(A),s) \\ &\qquad \wedge\ Holds(Clear(B),s] \end{aligned} \tag{1}$$

$$\begin{aligned} &Holds(InHand(A),Result(PickUp(A),s)) \wedge \\ &\quad Holds(Clear(B),Result(PickUp(A),s)) \\ &\quad \leftarrow Holds(Clear(A),s)\ \wedge\ Holds(On(A,B)) \end{aligned} \tag{2}$$

$$Inhand\ (A) \vee \exists B\ Holds(On(A,B),s) \leftrightarrow \neg\ Holds(Table(A),s) \tag{3}$$

We have omitted the cases of picking up a block which is stacked on the Table, or stacking a block on the Table, but they are easy to write. SC formulae such as these are the non-logical domain-dependent axioms. Formulae 1 and 2 are *effect axioms*; their antecedents are *pre-conditions* of the action, and their consequents are *post-conditions*. Formula 3 is a *domain constraint*, which is true for all situations (and therefore, time). It uses the idea that a fluent *Table(X)* encodes the information that block X is directly on the Table, in distinction to when it is on, e.g., block Y, encoded as *On(X,Y)*. In recent developments (see Shanahan, 1997 for a summary) the SC has been further

embellished with *causal* formulae or even predicates to capture the one-way nature of causal connectives, ie, toggling a switch causes the light to come on, but the light going off does not toggle the switch (as would unfortunately be the case if "causes" is encoded as implication). While we will not consider causal connectives or predicates, in Sub-Section 4.2 an account will be given of the apparent nondeterminism that can arise from a failure to represent causal information.

3. NONDETERMINISM

Informally, nondeterminism is simply the possibility of two or more distinct successor situations due to an action. There appear to be two ways to capture this. One is to modify the signature of the *Result* action to $Result : Act \times Sit \rightarrow 2^{Sit}$. This has difficulties arising from how one is to express what fluents hold in which alternative successor state, and would entail a radical re-formulation of the SC. We do not consider this way here. The usual way is to say something like the following formula, which informally says that one is uncertain which (or even both) fluent $\neg f$ or $\neg g$ holds in the successor state.

$$\begin{aligned} &[Holds(\neg f, Result(a,s)) \\ &\quad \vee Holds(\neg g, Result(a,s))] \\ &\leftarrow Holds(f,s) \wedge Holds(g,s) \end{aligned} \tag{4}$$

Figure 1 is a diagrammatic representation of four ways that axiom 4 could be interpreted. The two explicit circles in each sub-diagram represent a generic situation s and its next situation *Result(a,s))*. The expressions in each circle denote either the fluent literals that hold in the situation, or the possible combination of fluent literals that hold in it. For instance, the sub-diagram I reflects strictly what the Formula 4 says, ie, all that we know is that in *Result(a,s))* the *disjunction* $\neg f \wedge \neg g$ is true. It is what some call a *partial model*. However, if one desires full (or conventional ground) models, there are several possible refinements. II and III are more definite but still partial models. IV is one possible full model. We will remark on the differences below.

At this point it is appropriate to formalise the distinction between *true* and *apparent* determinism. First we observe that in SC the definition $Result : Act \times Sit \rightarrow Sit$ for the *Result* imposes determinism on all actions. So, strictly, if the system is to admit nondeterministic actions in the traditional sense in the theory of computation, we have to modify the signature to

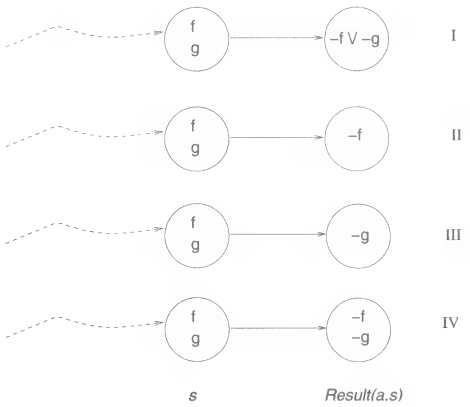


Figure 1: Three partial and one full model of Formula 4.

Result : $Act \times Sit \rightarrow 2^{Sit}$. Graphically, this would mean, for instance, that partial model I may be replaced by one shown in Figure 2.

The mere presence of an axiom of this form is not sufficient to ensure true nondeterminism of action a because there may be subtle interactions of the Formula 4 with the rest of the domain axioms. Let us denote the domain axioms collectively as Δ. For a given action a, let *ResultFormula(Result(a,s))* be the conjunction of all its post-conditions (possibly across several formulae). Definition 1 was introduced by Zhang and Foo (1996) to reduce nondeterminism to the bare minimum.

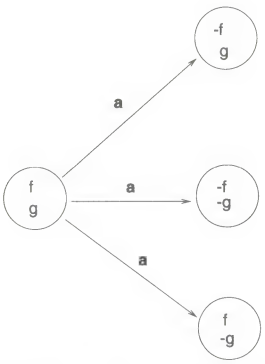


Figure 2: A true nondeterministic model of Formula 4.

Definition 1 *Let* $l_1 \ldots l_n$ *be fluent literals. If for all s,*
$\Delta \cup$ *ResultFormula(Result(a,s))* $\models l_1 \vee \ldots l_n$ *implies*
$\Delta \cup$ *ResultFormula(Result(a,s))* $\models l_i$ *for some i,* $1 \leq i \leq n$*, we say that action a is definite.*

We may call this the *subsumption condition*. Its role is so pervasive in simplifying theories of action that one is tempted to identify *definite* with *deterministic*. For instance, the description "definite" is used for Horn clauses because of its "unique head atom" condition. There are also the fundamental results from the simplest of nonmonotonic closures – the Closed World Assumption (CWA) (Shanahan, 1997). Recall that the CWA applied to a theory Γ is defined as $CWA(\Gamma) = Cn[\Gamma \cup \{\neg f \mid f \in Flu \wedge \Gamma \not\models f\}]$. Here Cn is first-order consequence closure. The CWA is "negation as failure" (applied to atoms) formalised and generalised. The "subsumption theorem" for the CWA is the following. Assuming that Γ is consistent, $CWA(\Gamma)$ is consistent iff for all atoms $f_1 \vee \ldots f_n$, $\Gamma \models f_1 \vee \ldots f_n$ implies $\Gamma \models f_i$ for some i, $1 \leq i \leq n$. Analagous to definition 1, one may say that the CWA is consistent iff it is applied to a (consistent) definite theory. This analogy is quite strong, as the subsumption condition in a clausal theory is equivalent to the existence of a least Herbrand model. And pure Horn clauses satisfy this condition. So does definiteness do the trick of eliminating nondeterminism? The answer is "yes", but it may need some nonmonotonic assumptions.

To return to Figure 1, if the action a is definite, then model I is not the best interpretation of the axiom 4 as it is not the most informative. Models II, III and IV are all possible. Observe that IV is a *completion* of both II and III in the sense that the situation *Result(a,s)* is completely specified, assuming that $Flu = \{f,g\}$ and that the rest of the domain axioms permit the completion. II and III are incomplete, and can be completed differently from IV – II by letting the remaining fluent be g, and III by letting it be f (again the domain axioms permitting). Confining attention to just the model

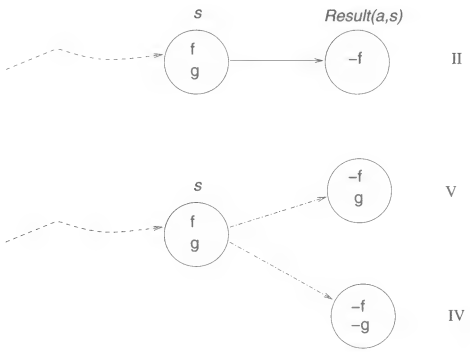


Figure 3: Apparent nondeterminism despite definite action.

II for example, Figure 3 shows how the two completions of II can be informally regarded as traditional (automaton-theoretic) nondeterministic transitions. One of these corresponds to model IV above; the other is the alternative completion which is now called V. This interpretation captures the "uncertainty" of the successor situation. This was in fact made explicit above in Figure 2, where the action *a* has a target which is the *set* of three situations.

The moral is this: definiteness (and correspondingly subsumption) is not sufficient to guarantee determinism. A natural way to eliminate one of the two possibilities IV or V in Figure 3 is the nonmonotonic assumption called *Inertia*.

Definition 2 *Let Δ and ResultFormula(Result(a,s)) be as in definition 1 above. Define Post(a,s) = $\{L \mid L \in Lit \wedge \Delta \cup ResultFormula(Result(a,s)) \models L\}$.*

Definition 3 *A domain is inertial if for all actions a, situations s, and fluent literal L, if $L \notin Post(a,s)$ then Holds(L,s) $\rightarrow$ Holds (L,Result(a,s)).*

Proposition 1 *If a domain is inertial and all actions are definite, then the domain is deterministic.*

Inertia is a very strong property. For instance, Foo and Peppas (2001) show that it implies the Markov property (which we will define below), which is informally the notion that the current state of the system as represented by its collection of fluent literals suffices to determine future states due to action sequences.

4. THREE SOURCES OF NONDETERMINISM

It is a deep philosophical question whether nondeterminism is inherent in some *macroscopic* domains, or is it just a representational convenience, not to say an artefact. Of course at the quantum level it is known to be "real". We do not presume to address the deep question. Instead, we will merely highlight a few system-theoretic issues at the level of domain encodings and representation that may not be familiar to the community but deserve some attention. There is a deep-seated resistance to nondeterminism as ontological reality to the extent that even Einstein refused to accept that "God plays dice". He and a number of similar minded physicists believed that the nondeterminism of quantum theory could be explained away at a deeper level by the introduction of *hidden variables* (now known not to be true). The same idea could ostensibly be introduced at

the macroscopic level in action domains that AI normally addresses to get rid of nondeterminism. Conversely, apparent nondeterminism arises when information is missing. The following sub-sections examine several kinds of missing information that may be reponsible.

4.1 The Markov Property

Consider the binary predicate *Equiv*: *Sit* × *Sit* defined by:

$$Equiv(s1,s2) \leftrightarrow \forall f\,[Holds(f,s1) \leftrightarrow Holds(f,s2)] \tag{5}$$

Definition 4 *A domain (or its theory) Δ has the Markov property (or simply, is Markovian) if for all actions a and all situations s1 and s2 ,*
$\Delta \models Equiv(Result(a,s1),Result(a,s2)) \leftarrow Equiv(s1,s2)$.

A Markovian system is often said to be "memoryless", in the sense that how it arrived at a situation s is immaterial once the complete set of fluent literals that hold in it is known. It is an easy corollary of the Definition 4 that if $s1$ and $s2$ have the same set of complete fluent literals, then their *future trajectories* must agree up to fluent literal equality (this is input-output equivalence (see Arbib, 1969a)).

What is less familiar is that any *k-memory* system can be transformed to a Markovian system. A k-memory system is one whose result transition (and therefore output) depends on the current action, and the last k situations that were visited rather than just the most recent one. To describe such a transformation, we recall the definition of an automaton (see Arbib, 1969a; Booth, 1968) and show a mapping of a k-memory automaton to an ordinary Markovian one. An (ordinary Markovian) automaton $\mathcal{M}$ is a quintuple $\langle I,Q,Y,\delta,\lambda\rangle$ where I is the set of inputs, Y is the set of outputs, Q is the state set, $\delta : Q \times I \rightarrow Q$ is the state transition function, and $\lambda : Q \rightarrow Y$ is the output function. A k-memory automaton $\mathcal{M}(k)$ is also a quintuple $\langle I',Q',Y',\delta',\lambda'\rangle$ with the difference being that the transition function is δ': $Q'^k \times I \rightarrow Q'$ instead. The detailed structure of Q'^k here is meant to be "the last k steps", captured by the formal definition of $\delta'..0$. A schematic representation of this is shown in Figure 4.

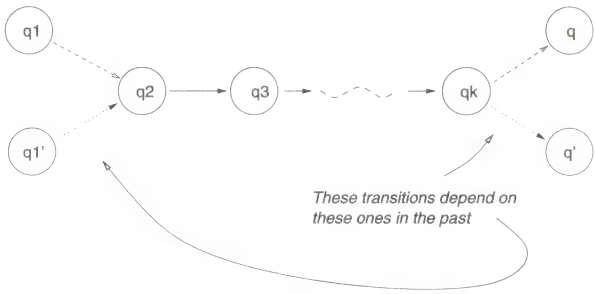


Figure 4: An example of a k -memory transition

In this figure, all actions except possibly the first one, are identical. The observer is assumed to see only the action sequence starting from $q2$, and then observes an apparent nondeterminism of the action on qk.The point is that the history of where qk came from does matter. If the starting point of the trajectory is $q1$, the next state is q; if it is $q1'$, then the next state is q'. In general for a k-memory system this dependence could be on any subsequence of the k-trajectory. The transformation of such systems to a Markovian automaton is almost obvious. We merely identify

the Markov states of M with *k-tuples* in $M(k)$, ie, each state in M is now $\langle q_1, \ldots, q_k \rangle$, and then define δ to map from inputs and such k-tuples to k-tuples (they share duplicate information, but this does no harm). So, in principle k-memory systems are not more expressive than Markovian systems.

The problem is that if we *represent* the k-memory system unwittingly as a Markovian system (ie, without first using the transformation above) then it will *appear* to be nondeterministic, as indicated in Figure 4 if one were to just observe what happens to the state qk with no knowledge of its past trajectory. The "hidden variable" here is that history. In fact, a standard scenario in the logic of actions, called the Stolen Car Problem (Baker, 1991), is exactly a manifestation of this history dependency. It can be described informally as follows. A car is parked on day 1; then 2 days pass, and on the third day it is noticed that the car is missing. Question: on which day was it stolen? On day 2 or day 3? There are two possible automata that represent the solutions, shown in Figure 5. The fluent f denotes the presence of the car and its negation $\neg f$ that the car is missing. The only "action" is the *Wait* action. The first automaton is non-Markovian – "history" matters. This dependence on history can be explicitly encoded using the transformation above, noting that it "remembers" how many times the action has been performed.

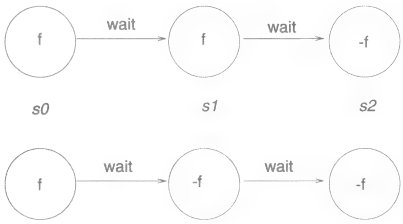


Figure 5: The Stolen Car Problem — Non-Markovian Systems

4.2 Causality

The failure to capture causal information is another common source of apparent nondeterminism. Perhaps the simplest example to illustrate this is one that is essentially due to Ginsberg and Smith, neatly summarised in Shanahan (1997). There is a room with two vents $V1$ and $V2$. If both vents are closed the room becomes stuffy. Using the obvious fluent representations, this can be written as a logical constraint

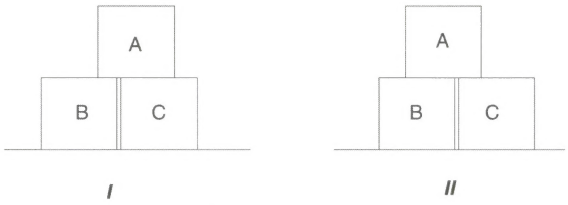
$$Stuffy \leftrightarrow (\neg V1 \wedge \neg V2) \tag{6}$$

Now, suppose we begin in a situation $\{\neg Stuffy, \neg V1, V2\}$, ie the room is not stuffy, vent $V1$ is closed and vent $V2$ is open. Then let us close the vent $V2$. Intuitively we would expect the resulting situation to be $\{Stuffy, \neg V1, \neg V2\}$. Unfortunately if we do not appeal to the intuition that *Stuffy* is what the engineers call a "controlled variable", and $V1$ and $V2$ are "controlling variables" (these have been re-invented in AI as "non-frame and frame fluents" (Shanahan, 1997)), there is no reason to exclude the alternative result $\{\neg Stuffy, V1, \neg V2\}$ in which the first vent magically opens to prevent stuffiness. Indeed, if one were to re-interpret the fluents in a different domain, the second alternative might be intuitive. The problem here is that what is really needed is a way to eliminate this unintended nondeterminism by re-expressing Formula 6 so that it captures the one-way direction of influence from cause to effect. There are many ways to do this, with Shanahan (1997) being a summary. A very recent approach by Thielscher (1999) combines constraints like 6 with

explicit *influence* information to encode the difference between controlling and controlled fluents. In many ways, the AI approaches to reducing nondeterminism have been anticipated by Zeigler, Praehofer and Kim (2000) in the area of modelling and simulation. There the notion of "influencer" and "influencee" is primary.

4.3 Parametised Actions

Sometimes apparent nondeterminism is occasioned by conflating parametised actions into a single one, an excessive abstraction. Perhaps the simplest example that demonstrates how this may lead to a hidden fluent is shown in Figure 6 where the two block configurations have exactly the same description if the only predicates in the language are *On(X,Y)* and *Table(X)* as usual (ignoring robot arms). The constraints are identical. However, if block *B* is removed, the two resulting situations will be different – in II block *A* will topple, but not in I. The problem is of course that the *Stack* action should have a parameter that expresses the skew in the stacking. Intuitively we know that there is a missing fluent that encodes the notion of *support*, ie we should have distinguished I from II perhaps by adding $support(A,C) \wedge \neg\ support(A,B)$ to the description of I, and $\neg\ support(A,C) \wedge support(A,B)$ to that of II. We are of course then obliged to adjust the rest of the theory accordingly. And yet, if indeed the language is restricted as above, both I and II are described by a common axiom set, say Δ . In that case nondeterminism will follow in this manner. Let α be the formula $Holds(On(A,C), Result(Remove(B,s0)))$ which says that block *A* has not toppled. Thus $\neg\ \alpha$ says that it has. Hence, to accommodate the two results, $\Delta \models \neg\ \alpha \vee \alpha$ which is nondeterministic. Further, it is not possible that either $\Delta \models \alpha$ or $\Delta \models \neg\ \alpha$, for if so Δ will fail to correctly account for either I or II after the Remove action (both are impossible if Δ is consistent). Hence any correct formulation of the domain with skew stacking cannot be based on this restricted language alone. In fact, this kind of argument can be more rigorously demonstrated with the aid of the Robinson Consistency Theorem (Shoenfield, 1967) which says that if the union $T_1 \cup T_2$ of two theories T_1 T_2 is inconsistent, then there is σ such that $\sigma \in T_1$ and $\sigma \notin T_2$, a topic addressed by Kwok (1998) where he subsumes hidden fluents under the more general "unobservable" or "theoretical" terms. We must however caution that even when we know that the introduction of hidden fluents will eliminate nondeterminism, it may take much ingenuity to discover what fluents to introduce, and how they fit into the existing partial theory. Space limitations do not permit us to discuss the obvious connection of this to machine learning.



What happens if block B is removed?

Figure 6: A Missing Fluent.

5. CONCLUSION AND ACKNOWLEDGEMENT

Three sources of nondeterminism were reviewed and explored. While nondeterminism is often a useful way to specify actions, for practical reasoning and implementations of systems it may be necessary to apply some of the nondeterminism reduction ideas that are discussed here.

The first author gratefully acknowledges the influence of Emeritus Professor John Makepeace Bennett. It was in the course of many conversations with Professor Bennett that the author learned the importance of asking the right questions, and the virtue of patience, persistence and integrity in research. The international and collaborative character of the author's research is a fulfillment of Professor Bennett's advice: (1) "It is no use being clever in private", and (ii) "It is a lot easier to fool yourself than to fool clever friends". The second author completed both his first and second degree under the tutelage of academic staff attracted to Professor Bennett's scholarship and ready wit. The third author was a beneficiary of the research environment that Professor Bennett provided for international students and visitors.

REFERENCES

ARBIB, M. (1969a): Theories of abstract automata. Prentice-Hall.

BAKER, A. (1991): Nonmonotonic reasoning in the framework of the situation calculus. *Artificial Intelligence* 49:5-23.

BOOTH, T. (1968): Sequential machines and automata theory. John Wiley.

FOO, N. and PEPPAS, P. (2001): System properties of action theories. *Proc. of the Workshop on Nonmonotonic Reasoning and Action, 17th International Joint Conference on Artificial Intelligence*, Seattle.

KWOK, R. B-H. (1998): Formalising aspects of theoretical term usage. PhD thesis. University of Sydney. http://www.cse.unsw.edu.au/~ksg/Abstracts/Thesis/rkwok.PhD.html.

SHANAHAN, M. (1997): Solving the frame problem: a mathematical investigation of the commonsense law of inertia. MIT Press.

SCHOENFIELD, J.R. (1967): Mathematical logic. Addison-Wesley.

THIELSCHER, M. (1999): From situation calculus to fluent calculus. *Artificial Intelligence* 111: 277-299.

ZEIGLER, B.P., PRAEHOFER, H. and KIM, T.G. (2000): Theory of modeling and simulation :integrating discrete event and continuous complex dynamic systems, 2nd ed. Academic Press.

ZHANG, Y. and FOO, N. (1996): Deriving invariants and constraints from action theories. *Fundamenta Informaticea* 30:23-41.

BIOGRAPHICAL NOTES

Norman Foo is a Professor of Knowledge Systems Group in the Department of Artificial Intelligence in the School of Computer Science and Engineering at the University of New South Wales. He was previously Professor of Knowledge Systems in the Basser Department of Computer Science at the University of Sydney. He holds a BE and ME from the University of Christchurch and a PhD from the University of Ann Arbor, Michigan. His research interests include knowledge representation, belief revision, non-monotonic reasoning, logics of action, diagrammatic reasoning, abductive reasoning, logic programming, systems theory and ontologies.

Rex Kwok is a Research Associate in the School of Computer Science and Engineering at the University of New South Wales. He has a PhD from the University of Sydney. He has eclectic interests including knowledge acquisition, logic formalisms and action results.

Yan Zhang is a Senior Lecturer in the School of Computing and Information Technology at the University of Western Sydney. He has a BSc and MSc from the University of Wuhan and a PhD from the University of Sydney. His interests include knowledge representation, nonmonotonic reasoning, logic programming and computational complexity.

Submission Guidelines

The Journal encourages submission of innovative and original articles in all areas of Information Technology including Computer Science, Software Engineering, Information Systems, Computer Systems and Information Engineering and Telecommunications.

The following forms of article are published:

- Full articles examining original research and/or application;
- Short articles describing original research and/or application or commenting on relevant legal, political or technical innovations;
- Survey or tutorial articles describing new directions in Information Technology or which are relevant to the practical application of fundamental research;
- Book Reviews;
- Announcements; and
- News Briefs.

Full details can be found at the Journal home page: www.jrpit.acs.org.au

Subscription Guidelines

The annual subscription for the Journal is

Individuals	
Australian Computer Society members	Free
Delivery address in Australia, incl. postage and GST when applicable	A$66
Overseas, incl. postage.	US$54
Institutions	
Australia, including postage and GST where and when applicable.	A$99
Overseas, including postage	US$81

All subscriptions to the Journal are payable in advance and should be sent (in Australian Currency) to the address below.

Advertising Information

The Journal accepts advertisements in the following categories:

- IT-related career opportunities
- IT-oriented training and education courses;
- IT-related conferences and workshops.

Rates are as follows:

Full Page	$1,500
Half Page	$750
Quarter Page	$375
By line	$100 for heading plus up to six lines of text (40 characters per line).
	$50 for each additional three lines, or part thereof.

Full details can be found at the Journal home page http://www.jrpit.acs.org.au

Bibliographic Information

Journal of Research and Practice in Information Technology – *J.Res.Prac.Inf.Tech.*
formerly the Australian Computer Journal (ISSN 004-8917)

ISSN	1443-458X
Circulation	15,500
Publisher	Australian Computer Society Inc. PO Box Q534, QVB Post Office Sydney 1230, New South Wales AUSTRALIA
Production Management	Associated Business Publications Pty. Ltd. P.O. Box 3267, Umina Beach, NSW 2257, Australia
Printer	Mailing and Print Services Pty Ltd 8 Aquatic Place, Frenchs Forest, NSW 2086, Australia
Cover Design	Modern Planet Design (08) 8340 1361 • modern@webmedia.com.au

SPECIAL ISSUE
A TRIBUTE TO JOHN BENNETT

Contents

273 John Makepeace Bennett – An Inspiration
Jenny Edwards

280 Early Contributions to IT in Australia, from the Basser Department of Computer Science – John Bennett's SILLIAC Years.
B. de Ferranti and J. Deane

294 A Network Flow Method for Determining the Gradient of Roads
A. Land

298 On the Complete Pivoting Conjecture for Hadamard Matrices of Small Orders
C. Koukouvinos, E. Lappas, M. Mitrouli and Jennifer Seberry

310 Waltzing on a KDF9
Don Herbison-Evans

315 Screw-Based Computer Graphics
I. Parkin

321 Strong and Weak Memory Consistencies
Chung-Kwong Yuen

325 A Logic Mini Trainer
Allan G. Bromley

329 A Sorry Tale – A Study of Women's Participation in IT Higher Education in Australia
Jenny Edwards and Judy Kay

336 Issues in the Deployment of Digital TV on the Internet
Chris Bennett

346 Broadband Internet Access in Regional Australia
Arthur Sale

356 B-Trees – A Simple Proof of Their Space Utilisation
J. Hext

360 Apparent Nondeterminism in Actions
Norman Foo, Rex Kwok and Yan Zhang

A publication of the Australian Computer Society
ISSN 1443- 458X